The Autodesk File:

Bits of History, Words of Experience

Edited by John Walker

Except as otherwise noted in the text, this book is in the public domain and may be reproduced, distributed, and quoted as long as the source is cited. The Third Edition of this book was published by New Riders Publishing in 1989 as ISBN 0-934045-63-6, with copyright retained by Autodesk, Inc. This edition was prepared solely by the editor, to whom Autodesk returned rights to all the material in the Third Edition plus the new Autodesk documents which appear here. "Autodesk" is used herein solely to identify the company, not to imply endorsement of or responsibility for the content of this book. Autodesk retains no copyright interest in the material published herein, and is neither responsible nor liable in any form for this book and its content.

Fourth edition. Version 4.0, March 1, 1994.

This book was published in February of 1994. Camera-ready copy was produced with the LaTeX document preparation system, developed by Leslie Lamport.

Autodesk, the Autodesk logo, AutoCAD, AutoCAD AEC, AutoSketch, AutoLISP, and Authorized AutoCAD Training Center are registered trademarks of Autodesk, Inc. AutoSolid, AutoShade, CAD/camera, The Engineer Works, ADI, DXF, ACAD, Advanced User Interface, AUI, AutoFlix, Auto Book, AutoCAD-80, AutoCAD-86, and Autodesk Forum are trademarks of Autodesk, Inc. IBM, PC/XT/AT, Personal System/2, and PC-DOS are trademarks of International Business Machines Corporation. Apollo, DOMAIN, and AEGIS are trademarks of Apollo Computer, Incorporated. DEC, VMS, VAX, and VAXstation are trademarks of Digital Equipment Corporation. Sun Microsystems, Sun Workstation, Sun–2, and Sun–3 are trademarks of Sun Microsystems, Incorporated. Microsoft, MS, MS-DOS, and XENIX are registered trademarks of Microsoft Corporation. Intel, 8080, 8086, and 8088 are trademarks of Intel Corporation. Lotus, 1-2-3, and VisiCalc are trademarks of Lotus Development Corporation. Texas Instruments is a registered trademark of Texas Instruments Corporation. WordStar is a registered trademark of MicroPro International Corporation. BASIC is a registered trademark of the Trustees of Dartmouth College. dBase, dBase III, and Ashton-Tate are registered trademarks of Ashton-Tate. Unix is a trademark of AT&T Bell Laboratories. TEX is a trademark of the American Mathematical Society. CompuServe is a trademark of H&R Block, Incorporated. Marinchip, M9900, QBASIC, and WINDOW are trademarks of Marinchip Systems, Ltd. The Golden Age of Engineering, Tools For The Golden Age Of Engineering, and Turbo Digital and its associated logotype are trademarks of Marinchip Systems. CP/M, CP/M-80, CB80, CB86, CP/M-86, and PL/I-80 are trademarks of Digital Research, Inc. Glasnost and Perestroika are trademarx of the Communist Party of the Soviet Union. Xanadu is a trademark of Xanadu Operating Company, Inc. Labitat is a trademark of External Tanks Corporation. Apple is a registered trademark of Apple Computer, Inc. Macintosh, Lisa, and HyperCard are trademarks of Apple Computer, Inc. Cray and X/MP are registered trademarks of Cray Research, Inc. COMPAQ is a trademark of COMPAQ Computer Corporation. ComputerLand is a trademark of ComputerLand Corporation. Selector V and Micro-Ap, Inc. are trademarks of Micro-Ap, Inc. ANSYS is a registered trademark of Swanson Analysis Systems, Inc. Post-it is a trademark of 3M Corporation. Bernoulli Box and the IOMEGA logo are registered trademarks of IOMEGA Corporation. MSC-NASTRAN is a registered trademark of the National Aeronautics and Space Administration. Z80 and Z8000 are registered trademarks of Zilog, Inc. Bic Mac is a trademark of McDonald's Corporation. SOLUTION 3000 and MILLMASTER are registered trademarks of Micro Engineering Solutions Inc. DesignPlus, ManufacturingPlus, DesignExpert, and ManufacturingExpert are trademarks of Micro Engineering Solutions Inc.

Other product names mentioned herein may also be trademarks used here for identification purposes only.

Contents

Introduction 10

Working Paper 13

Information Letter 1 31

Information Letter 2 43

What to Name the Company 47

Information Letter 3 49

Information Letter 4 57

Initial Stock Distribution 62

Information Letter 5 64

Information Letter 6 80

Information Letter 7 88

Information Letter 8 104

August 1982 Meeting 106

AutoCAD-80 Development Log 114

1982 Annual Meeting 139

Quality Department Priorities 143

Meetings: December ’82, January ’83 145

February 1983 Meeting 149

March 1983 Meeting 157

June 1983 Meeting 163

AutoCAD Wish List 171

Crisis Letter 185

John Walker’s Business Plan 189

Proposed Autodesk Organization 193

July 1983 Meeting 201

Low Rent 3D 207

Electric Malcolm 211

October 1983 Meeting 215

Piece of Cake 219

Information Letter 10 221

Information Letter 11 227

The Deal on the Table 233

AutoCAD Lite 240

Taxes and Such 244

Why Lisp? 250

Number One 252

Prime Time 254

Initial Public Offering 260

Surplus Value 274

Information Letter 12 280

Time of Turbulence 290

Speech at Silverado 296

Super Programmers 305

CAD: the Heart of Computer Science 307

The Computer Revolution 311

AutoBits 319

Flat-out Programming 343

Hardware Lock Debater's Guide 345

Information Letter 13 349

Cadetron and Solid Modeling 355

Removing the Hardware Lock 359

Urgent Fury 363

The Portable Data Base 372

Breaking the 3D Logjam 382

Jeremiad 385

Urgent Worry 389

The Golden Age of Engineering 392

Cosmic Perspective 394

External Tanks 396

Source Distribution 398

The Morning After 404

Glasnost 408

Where's It All Going? 412

Xanadu 416

Bored of Directors 420

Valedictory 422

L'Envoi: the First Six Years 428

Technological Leadership 430

Through the Looking Glass 439

AutoCAD Expo Moscow 453

Golden Hammer 455

One Hundred Million Dollars **457**

The New Technological Corporation **459**

Generic Software Acquisition **489**

Why Cellular Automata? **491**

Understanding AMIX **497**

Batting .300 **501**

We'll Return, After This Message **504**

Caustic Moment **512**

The Hardware Lock Strikes Back **524**

"Autodesk Arrogance" **530**

Why Molecular Modeling? **534**

Whither AutoCAD? **538**

Nanotechnology in Manufacturing **558**

Max Q **580**

Information Letter 14 **600**

View from the Ridge **646**

Valuing Corporate Image **653**

Simulation Isn't Reality **656**

The New Autodesk **659**

Choose Wisely 683

Added Value 690

Stating the Problem 691

Al Green Retires 697

Incentives, Deterrence, and Defence 700

Searching for a New CEO 705

European Software Centre 709

The Dark Night of the Soul 718

Reporter at Work 730

AutoCAD for Windows 781

MicroTimes 784

New Leadership for Autodesk 807

Farewell, Xanadu 810

Micro Engineering Solutions Acquisition 813

Patent Nonsense 819

Programs Are Programs 822

AutoCAD LT 832

Creation/Evolution 842

Ten Years After 846

Appendices **848**

A Before Autodesk **848**

B The View from Sweden **856**

C Autodesk Trivia Quiz **864**

D Financial Results **873**

E Autodesk Stock Price History **875**

F Product Release History **877**

INDEX **882**

Autodesk Founders. They didn't walk on water, but they could fly pretty well.

Introduction

How to begin to tell the story of Autodesk? The company was so unusual in its origin, so unconventional in its growth, and so eventful has been the road that started with a small group of programmers sitting around talking about building a company and has led, so far, to a multinational company which is the undisputed leader in its market, that it's tempting just to shrug your shoulders and say "you had to be there".

Because Autodesk started out as a very decentralised organisation and has remained one to some extent, and also because of the prolix proclivities of its founders, who would rather write a book than talk on a telephone for ten minutes, the genesis, evolution, and history of Autodesk has generated a large volume of paper.

Unlike many companies, whose history can be recovered, if at all, only by a major oral history effort, one can watch Autodesk develop by reading the documents that were, during the company's development, the primary means of communication between the people involved. Reading these documents lets you see how assumptions we seldom question today got cast into concrete, how many blind alleys we had to explore to find answers which seem, in retrospect, utterly obvious, and how throughout the history of the company, when a major effort was called for to advance the company, Autodesk people have always responded with the energy, creativity, responsibility, and dedication which are the largest reasons for Autodesk's great success in the market.

Too many business books, like histories of science, tend to tell the story as a straightforward progression from start to finish. Reality is never that easy. Decisions are made in the face of incomplete and unreliable information because they *must* be made. There's no way to tell a promising avenue of success from a blind alley when you turn onto it—you only find out much later. As you read through these documents, you'll be seeing it all, and if it seems tedious and repetitious, it's because the process of building a company *is* often tedious and repetitious. But it's also rewarding, and I hope that these documents also convey the feeling of exhilaration, challenge, and accomplishment that everybody felt as we built this company into what it is today.

When you read these documents, you're opening time capsules buried as Autodesk developed. The documents are presented with essentially no editing other than that required to convert them from the variety of document processors in which they were written into TEX. Some irrelevant material, such as five-year-old name and address lists, has been deleted but no elisions have been made which rewrite history, cover up errors, or otherwise alter the record. Where appropriate, I've added footnotes to explain matters which might not be clear at several years remove and to call out important items mentioned in passing in the text.

Since this is a history in documents, the picture of the company it presents is unavoidably coloured by the documents available when this history was prepared. The resulting collection weights my contribution heavier than it was because I write prolifically and keep everything I write. It covers AutoCAD-80 far out of proportion to its importance because the AutoCAD-80 logs exist in machine-readable form and the AutoCAD-86 logs do not. There is little coverage of the rich history of CAD/camera, and little of the development of Autodesk's marketing and sales organisation. The history is also weighted toward the early days of the company because as

the company has grown business has come to be transacted far more in meetings and via ephemeral memoranda than in explicit status reports. As a result, nothing of the second public stock offering has been included, nor anything of the development of AutoCAD AEC or of AutoSketch. The absence of documents in this history is simply the effect of what has been preserved, not an attempt on my part to emphasise or diminish the importance of any aspect of the company's development or any individual's contribution.

I don't know whether these documents show how to start and run a company that assures success in a treacherous market or whether they simply chronicle the education of a group who was in the right place at the right time. Probably nobody ever will know. But from a sample size of one, it's the only way I know to start a wildly successful company, and improbable as it may seem, this is how it really happened.[1]

Notes to the second[2] edition.

This second edition was made possible by the work of many people in digging up and in some cases typing in numerous documents which were not available when the first edition was prepared. In addition, the eagle-eyed proofreading of several people found many egregious errors and omissions in the first edition which, thanks to their efforts, have been corrected herein. For their exertions, special thanks are due to Peter Barnett, David Ciari, Lew Goldklang, Duff Kurland, Valerie Lowe, Steve McCall, and Lars Åke Moureau.

While many errors have been corrected and additional documents added, this volume remains a woefully incomplete account of the rich experience that living through these times has been. Perhaps in some placid period in the future I'll find the time to adequately relate such stories as "The XOR Patent",[3] "April 31, 1985", "Random Vectors", and "Get me to the SEC on time". For the moment these stories of the Hungry Rats of Autodesk must slumber alongside the Giant Rat of Sumatra, tantalising yet untold.

Notes to the fourth[4] edition.

In a way, all the earlier editions of *The Autodesk File* were incomplete. They chronicled the exhilarating and frequently exasperating experience of starting a company from nothing and seeing it grow into a leader of an industry it helped to create, but, written in the midst of ongoing runaway success, gave the impression that continued success was merely a matter of doing the same things as before, that entrepreneurship and leadership of an industry were one and the same. Indeed, in 1989, when the third edition of *The Autodesk File* (the New Riders "Purple Paper Eater" book) appeared, many people at Autodesk, myself included, *did* believe these things.

How naïve we were.

I originally began assembling this edition to commemorate Autodesk's tenth anniversary. I guess it's only appropriate, given the history you're about to read, that the tenth anniversary edition of *The Autodesk File* show up two years late and much, much larger than originally anticipated. The book has just about doubled in size from the 1988 edition, but then the company is twice as old today and a great many things have happened since

[1] This introduction was written for the original in-house edition, distributed in December of 1987.

[2] The second edition, circulated in-house, was distributed in June 1988. Notwithstanding its having been reviewed by both the legal and accounting departments beforehand, it was promptly stamped "Confidential" and made subject to a sign-out procedure. Stripped of trivial information, such as AutoCAD unit sales numbers and profit and loss by subsidiary (by then well out of date), it became the raw material for the edition published by New Riders.

[3] But see page 819.

[4] The third edition of *The Autodesk File* was published in 1989 by New Riders Publishing as ISBN 0-934035-63-6. That book is now out of print. The third edition was an abridged version of the in-house second edition, with chapters shuffled around and some narrative bridges added to tell the story around the documents. This book contains everything in the third edition (except the "bridge" sections), with all the material dropped in that book restored and more than 400 pages of new documents from the years 1988–1993.

then. A great, great many things.... Legally, corporations are people, but in reality they're very different from you and me. They don't get wrinkles in their brows from worrying, ulcers from stress, or go bald from ripping their corporate hair out in frustration. They age and become set in their ways, but rejuvenation is as close as the next person they hire, given the wisdom to listen and the courage to change. This is a story of birth, growth, maturity, aging, and rejuvenation. All of it is a story of change. The story remains incomplete. I hope it will remain forever incomplete, for the last word of the final chapter of a complete history must chronicle the end of this venture born with such hope in 1982. Corporations aren't people; with wisdom and courage, and yes the luck to find the right people at the right time, immortality can be theirs. Let us hope that is Autodesk's destiny, and strive to make it so.

John Walker

Neuchâtel, Switzerland
February, 1994.

Working Paper

The Working Paper was the document which resulted in the formation of Autodesk. I wrote it at a time when it was clear that Marinchip Systems, the company that I had started in 1977, and which Dan Drake and I had operated since 1980, did not have a bright future. In an attempt to find markets for Marinchip's software, we had been talking to the OEM division of Lifeboat Associates. It was on a trip with Lifeboat to computer companies in the Los Angeles area in December of 1981 that I first formed the idea of starting a software-only company to provide software for the coming tidal wave of small computers from large manufacturers. This working paper was written in 48 hours, after weeks of thinking about what to do. This paper served as the introduction of the concept and the invitation to the meeting to organise the company.

Marin Software Partners
Working Paper

by John Walker
Revision 4—January 12, 1982

Introduction

This document is a working paper which sets out the background, general business plan, and strategy of Marin Software Partners (MSP), a new company to be formed by some of those who read this paper. The major goal of Marin Software Partners will be to develop and market software packages, primarily application but also system, for popular mass-market computer systems, including, but not limited to, CP/M, IBM 8086 DOS, and Unix System III.

Background

Marinchip Systems and many of those associated with it in various capacities have discovered that while it is possible to earn a reasonable living attempting to be a full-service computer company through the massive exertion of effort and consumption of physical capital, it is not possible to achieve the success that has accrued to those who let the mass market do their selling for them. The possessor of a unique software package such

as Visi-Calc or Wordstar finds that much of the promotion of the package is done by the hardware vendor or systems house who wants to sell a system by providing the capability the package offers.

It is far too late in the game for a successful start-up of a full service computer company without massive venture capital and an organization which none of us knows how to manage. Furthermore, the chances of success against those with literally unlimited advertising budgets and marketing organizations (IBM, NEC, etc.) are very slim. However, the software business is very different. First of all, a software package can be produced out of pure effort, with only the capital needed to finance the machine and pay the programmer. Unlike hardware, the big vendors of mass market machines are mostly utterly ignorant regarding software, and software manufacturing is as easy as copying discs. In addition, independent software marketing channels such as Lifeboat Associates exist and are working in cooperation with major hardware vendors (Xerox, HP, Altos) to sell application software to purchasers of hardware systems.

I feel that at the present time it is possible to, albeit with high risk, start a software firm with the capital available from Marinchip Systems, and that this is the best possible deployment of that capital. No conceivable investment in the business of Marinchip has the probability of generating a comparable return. Unlike the hardware business, MSP will be in the middle tier of companies in its business, and will likely be in the front rank based on competence and professionalism.

Which brings me to...

The game has changed. In 1977 this business was *fun*—the sellers and buyers were hotshot techies like ourselves, everybody spoke the same language and knew what was going on, and technical excellence was recognised and rewarded. Today, the microcomputer industry is run by middle manager types who know far more about P/L statements than they do RAM organization. They are the people who determine whether you succeed or fail, and their evaluations are seldom based on technical qualities. Hence, the first thing any venture in this field has to be is businesslike.

What this means is that, first of all, any person who is unwilling to assign this venture a priority equal to or above his current employment does not belong in MSP. That doesn't mean you have to quit your job to join MSP. What it does mean is that if you say you agree to a certain share, then you will deliver that share week after week, month after month, year after year regardless of other commitments except in the case of total catastrophe which would cause you to equally neglect any other job you have. In working with people associated with Marinchip, the following conversation has occurred more than once:

> "When will it be done?"
> "Well, I don't know."
> "Why not?"
> "Well, I know I told you it would be done by now, but a lot of stuff came up at work and I..."
> "Isn't this work? Don't you get paid for it?"

If you view your work with microcomputers as a hobby, if you look on the microcomputer business as a way to write off your home computer on your taxes or mollify your spouse about the money you spend on computers, if you're looking for a supplementary income to pay for a disc drive or outboard motor or whatever, you do not belong in MSP. MSP will be composed exclusively of people who intend to develop quality products,

aggressively market them, and reap rewards far greater than those available from their current employment. We don't expect most people to start on a full-time basis; in fact, we're deliberately organizing the company to provide full time support services to moonlighting implementors, but if we're successful, we expect those involved to increase their commitment as the business grows.

If you feel, as I do, that a competent software person with the marketing connections to decide what to do and how to sell it is in the best possible position today to become very wealthy, then you belong in MSP.

General development strategy

Marinchip Systems has developed and is expanding a business relationship with Lifeboat Associates of New York City. Lifeboat is probably the largest independent software vendor in the world today, and is the primary source for application software for almost all the mass market computers sold currently. Through technical review of Marinchip products and presentations to Lifeboat customers, conversations with Lifeboat personnel, and negotiation of a very complex OEM agreement, Marinchip has come to be seen by Lifeboat as a competent organization in both the business and technical senses.

Lifeboat has expressed an interest in working with Marinchip to develop Marinchip products to be marketed through Lifeboat, particularly a QBASIC compiler for the 8086 (IBM) and Z-80 processors. Additionally, our contacts with Lifeboat give us the ability to sound out market demand for various packages, get tips on what people are asking for and not able to find, and also contacts with OEMs who want specialized work done.

Clearly then, one of the first tasks of MSP after formation will be to meet with Lifeboat and explain our business plan to them and get feedback and suggestions. I think that we already have the credibility to get work funneled our way by Lifeboat, and in any case the contacts are invaluable for market research.

MSP will concentrate on development of specific products with clearly defined functions. We will not attempt to implement grandiose systems and will not stray too far into the systems programming arena. Any program we develop must require little or no customization for installation, and little or no user consultation after sale. Otherwise, we can't afford to sell it. We're aiming for packages like Visi-Calc, Selector, Supersort, Wordstar, etc.

MSP must budget a substantial percentage of its capital for advertising and promotion. Undoubtedly, some packages will be largely marketed for us, but we cannot assume this and must realize that a market must first be created through advertising before it can be sold to.

Form of organization

MSP will be organized as a partnership. The general partner will be Marinchip Systems Ltd. (MSL), and the limited partners will be all the individuals associated with the company. Using this form of organization provides the limited partners the limited liability of a corporation without the disadvantages of double taxation of earnings, the risk of royalty income causing the corporation to be construed as a "Personal Holding Company" subject to 70% punitive tax, and the general hassles of operating a corporation.

MSL, as general partner, will be responsible for the day-to-day operation of the company. It will provide the

following services:

Headquarters services. Phone answering, order taking, shipping and receiving, copying.

Administrative services. Accounting, banking, billing, A/R maintenance, preparation of reports to partners.

Marketing services. Contact and negotiation with Lifeboat and other distribution channels, ad agency interface and copy preparation, ad placement, trade show exhibition. Market research and potential product evaluation.

Project coordination. Central message dispatching between partners. Monitoring of project schedules and reminders of delivery dates. Follow-up of customer complaints and suggestions.

Manufacturing. Manual printing, disc copying, inventory maintenance.

Limited partners will be responsible for the following:

Product development. Design, implementation, and documentation of new products.

Product maintenance. Correction of reported problems, adapting existing products to new hardware/software systems, installation of new features, revision of documentation.

Product evaluation. Pre-sale evaluation of products developed by other partners, preparation of critiques and problem reports for those products, interface with other partners in correcting those problems. Evaluation of competitive products from other manufacturers, preparation of reports on those products and selection of features and capabilities for incorporation in our own products.

Market research. Review of new product announcements, news items, advertising, and product demonstrations with an eye to potential new markets, competition, and opportunities. Preparation of summaries of important items for distribution to other partners.

Marketing assistance. Preparation of new product announcements, skeleton ad copy, and product brochure copy. Attendance at shows and at meetings with customers. Telephone consultation with important customers and potential customers.

Planning assistance. Participation in regular partnership meetings. Assistance in evaluation of partnership goals and new product selection. Technical assistance to other partners in areas of specialization.

Mode of operation

MSP is intended to be a "tightly coupled" business venture. It *is not* a front for marketing individual products and funneling royalties back to implementors. It is a partnership where partnership profits are distributed to partners based on their percentage ownership regardless of their source. Why? First of all, one of the major reasons to form a partnership rather than just going off on your own is the potential synergy of the various partners and the work they develop. We hope to offer software components which can be used together in meaningful ways, and as we go, to accumulate a "bag of tricks" (e.g. screen formatting routines, database access utilities, etc.) which make development of new products by all partners easier. If each partner were essentially on his own, we could easily spend more time figuring out cross licensing and royalties for shared components than in actual development. It would force any partner to evaluate, for each potential component used, the tradeoff of paying for it or doing it over. This is silly and counterproductive.

Secondly, it enables us to cut the risk to each partner while remaining able to swing our resources behind those products which "take off". Assume we develop five products and four are losers or barely break even but one becomes the "next Visi-Calc". In the "royalty payback" company we would have four unhappy implementors and one fellow with a rapidly increasing bank balance but the inability to adequately follow up the initial product with follow-on enhancements and adaptations. With the true partnership, we can commit our resources to a successful product as its success requires so that we can not only make a splash with it, but aggressively follow up the initial success with the new versions, new machine implementations, and additional features needed to expand and preserve market share.

I view the difference between the lone wolf implementor and the software marketing partnership as the difference between gambling and business. The lone wolf has the possibility of a higher return, but far less probability of realizing it. What matters in business is to be able to fail a large percentage of the time and still come out ahead. Having had several blockbuster products and having watched them diddled away by insufficient promotion and inability to concentrate resources on them as they showed promise convinces me of the truth of this statement.

Once MSP commences operations, we will select a set of products to develop and formulate, in advance, a development schedule, marketing plan, marketing budget, and cash flow projection per product. MSP accounting will be structured so as to produce actual figures on a monthly basis which update the projection. We will have partnership meetings on a monthly basis (or more frequently) in which each active project is reviewed from a technical and marketing standpoint and a decision will be made to continue, drop, or increase commitment to the project. Each new product we choose to undertake will be formulated and managed this way, so we are constantly forced to target the very limited resources we have on the segments of our business which are developing well.

Partners in MSP will prosper as the company as a whole does. This may help them to better evaluate products and projects based on their actual prospects rather than an attachment to something based on the amount of work that has gone into it or an attraction to an idea because it seems good. Our goal is to be able to react rapidly when a product takes off and build other products around it.

It doesn't take a lengthy look at the computer industry to conclude that the products that succeed are not always the best ones. Arguing with the marketplace may make you feel good, but it's about as productive as standing on the tracks and arguing with the Twentieth Century Limited. One of our chief goals in structuring the company is to promote rapid feedback of real-world information into the decision making process. I know how important this is—any reasonably dispassionate analysis of Marinchip's business would have concluded

as early as 1979 that the 9900 was a dead end. Yet the seductive lure of the "previous investment trap[5]" was such that two more years of effort were poured down a hole whose prospects of return were very limited.

That's not to say that having long term goals isn't important or that you should have no time horizon beyond the next month. There's nothing wrong with a slowly developing business with a large prospect of deferred return as long as it doesn't bleed your resources and result in your going under just when the world realizes that it needs what you've been selling for the last five years. What we have to guard against is blindness to a competitive idea (for example, screen-oriented word processors) which is sewing up the market while we still try to push something time has left behind.

Product development cycle

The business of MSP will be structured around products. Each product will be clearly defined and a written plan will exist for each product. At any given time, it will be possible to list all the active products and review their performance.

Each product will follow a well-defined life cycle. It begins when somebody decides that something looks like a good potential product. This is briefly written up and then discussed at the next planning meeting. If the product looks like it might be worth doing, one or more partners undertake the preparation of a development plan. The development plan spells out the specifications for the final product (at the level of detail a brochure might offer), lists potential competitive products and why ours would be better for the potential purchaser, and estimates the time and other resources which would be required for development. If after reviewing this plan the product still looks good, we sound out potential marketing channels and supplement the plan with projections for marketing cost and sales. The final plan is subject to approval by the partnership before development is started. Once development is authorized, the project goes into the implementation phase.

During the implementation phase, the partner or partners responsible for the project write and test the code and prepare the user manual. Those responsible should be left alone as much as possible during this phase. Only a devastating competitive announcement should be reason to reopen the project for consideration while implementation is underway. As long as it is on schedule, the project is of little concern to the other partners. Once an initial version is completed, including documentation, the project moves to the evaluation phase.

In the evaluation phase, a completed user copy of the package is given to a partner who has little knowledge of its internals and is in a good position to evaluate the package from a user standpoint. That partner's critique of the package as well as bug reports from the initial testing are used to refine the package so that the first release meets the highest professional standards. Remember, outfits like Lifeboat evaluate a package based on their customers' first impression of it. A rough first release can doom the package's prospects. While this evaluation is going on, the manual is edited into final camera ready form, advertising copy is prepared, and product brochures and other promotional material are prepared and printed. When the package has been shaken down to the extent that all are happy with it, it moves to the initial marketing phase.

In the initial marketing phase, manuals are printed so that orders can be filled immediately. New product announcements are sent to all trade publications and advertisements are placed as specified in the plan. Marketing channels (e.g. Lifeboat, etc.) are contacted and provided with sample copies, presentations, and/or demonstrations of the package. If trade journal articles have been prepared about the package, they should be timed to appear during this time. We want to have the maximum impact possible with the introduction of the package

[5] As defined in *How I Found Freedom In an Unfree World* by Harry Browne, Avon, 1973, Page 136.

to prompt people to try it. After they try it, we hope the package will sell itself on its merits. This is the phase in which the largest negative cash flow will be experienced, and the project will be constantly reviewed against the plan to make sure it is within the budget. As orders begin to come in, the negative cash flow begins to turn positive and to pay back the initial marketing debt. As this happens, the project moves to the marketing follow-up phase.

In the marketing follow-up phase, we find out how well we've done. The project is reviewed based on:

- Sales
- Cost of support
- User comments
- Dealer comments
- Competitive developments

and based on those considerations, we decide how to treat the package. We want to be as responsive to bug reports as possible, and to regularly release updates and enhancements. We want the user to feel that the package is "alive", not a take it or leave it item. Also, we develop a profitable aftermarket in updates among those already committed to the package. As long as a project is still active, we budget funds for advertising and other marketing, and our goal is to pyramid the success of products which sell well. This means (*and this is critical*) that our first priority is support, enhancement, and promotion of those products which are doing well. We don't know in advance which of our products that (or those) will be—we have to let the market tell us, but we have to listen and respond to the market's message. Marinchip's greatest failure was to develop a product and then not follow it up because another attractive development project was dreamed up. We cannot let that happen here.

Optimally, the success of one or two of our products will lead to natural follow-on projects (as Wordstar led to Mailmerge, Spellstar, etc.), which build on the sales of the original product (to start with, users of our first product are very likely to buy the add-on). That way we can let the market lead us into the area of business we do best in. We should review new product proposals in the light of our existing products, to see whether they complement them. Not that we shouldn't enter new lines of business, but those companies that have succeeded have done so by concentration, not by breadth of product line.

If a product fails to meet its sales plan, then in the follow-up we will review its performance and the reasons for its failure. Based on this review, we may decide to terminate the project or to remedy the product based on market response or to modify the promotion campaign based on reactions received. However, we must avoid throwing good money after bad, and we should expect a majority of products to fail and their projects to be terminated. That's why we establish an advertising budget in advance and stick to it. Only exceptional and well documented changes in the marketing environment should cause us to decide to increase our potential loss on an unsuccessful project.

Obviously the time scale of all of this will depend on the magnitude of the product undertaken. It's conceivable that a little CP/M utility might go from concept to follow-up in 2 months (although advertising lead times would limit the impact of advertising until later). Given the resources we have, I don't think we should undertake any project where the follow-up comes any later than 9 months after the project is first defined. We just aren't rich enough to piss away our resources for longer than that on a potential loser. If we decide that we want to do a massive system with lots of parts, let's do it in pieces that are individually salable. Then we not only get user feedback to guide our future development, that development is paid for from sales revenues, not from our pockets.

Money and management

Capital for the formation of MSP will be contributed by the general partner (MSL) and the limited partners. Partnership interests will be calculated based on the percentage of capital contributed to the initial capitalization of the venture. The law requires the following:

Limited partners cannot purchase their partnership interests through contribution of services [Dible[6] P. 180], but must contribute tangible assets. Management and operation of the company is solely the responsibility of the general partner (MSL). Violation of these rules either invalidates a partner's ownership share or exposes the limited partners to potentially unlimited risk in case of failure of the business, lawsuit, etc.

We do not want to select potential partners in this venture based on their bank balances, but rather their competence, willingness to work, and entrepreneurial orientation. However, we don't want to give away partnership interests or make participation a no-risk venture for any partner. The owners of MSL are basically risking everything they've made for the last 5 years on this venture; the amount of money we intend to contribute would let us lie on the beach for a long time, and we intend to make a lot more than we contributed to compensate us for the risk, the work, and the hard times ahead. We want to know that our partners in this venture have a stake in its success at least proportional to their ownership of the company.

The following plan is suggested for initial capitalization of the company: we will calculate the desired capitalization and the partnership shares of all partners. As noted above, partnership shares will be in direct proportion to contributions. Partners may purchase their shares either in cash, by a no-interest loan from MSL secured by equipment, or by a regular market-rate callable loan from MSL.

Here's how it works. Suppose a partner wants to buy in for $5000. The simplest thing is just to pay the $5000 in cash. Alternatively, since many partners will want to purchase machines for software development or already own them, they may use the money to buy a machine (getting the tax credit and depreciation benefits, which are incredibly attractive today), then pledge that machine as security on a zero-interest loan from MSL. Or, MSL can loan the partner the money on a regular unsecured loan at market interest rates, and that money can be used to buy a partnership share in the normal way. At, say, 20% you can "rent" $5000 for $1000 per year.

The idea of all this is that we recognize that a substantial portion of the initial capitalization is going to be used to buy machines for software development. Those partners who already own machines should not be forced to subsidize those who haven't, nor should those partners who obtain machines for MSP work be forced to forgo the tax benefits of buying the machine themselves. By loaning at no interest against the machine, we're allowing machine investments to be applied to partnership share dollar for dollar.

On all of these loans, it will be part of the agreement that revenues from a partner will first be applied to retiring any debts to MSL, and only then will the partner be paid directly.

Note that none of the above has been reviewed in detail for possible adverse tax consequences (in particular "imputed interest") and it's possible that there may be some more tax-attractive way to go at this involving leasing. Externally, this venture looks very much like a tax shelter, so the tax ground is very carefully covered and one must tread with caution in possibly questionable areas.

It should be clear that if MSL loans a partner the money to buy in, that loan should have an equal position in the recipient's mind with a home mortgage or auto loan. It is a real loan of real dollars which could have otherwise been spent by the principals of MSL on themselves. It is not "funny money" or a paper accounting

[6]Donald M. Dible, *Up your OWN Organization*, Entrepreneur Press, 1974.

transaction, and he who receives it should expect to pay it back, hopefully from revenues of the products MSP sells, but from other sources in the event MSP fails. Not only is this a realistic representation of what's really going on, it will hopefully inspire in all partners the kind of seriousness about this venture with which MSL approaches it.

If MSP fails, I will lose everything I've made for all the work I've done since 1977. I want partners who are willing to work as hard for success as I am.

Legally, limited partners have no say in the operation of the business. It is our intent that the business will be run as any other partnership based on partnership interests. Since I expect MSL to hold a controlling interest, this will probably make no practical difference. I believe that the people involved in this venture should be compatible enough that consensus will govern most actions taken by the partnership. This business can succeed only if all partners work to make it succeed. Since MSL has the most to lose, MSL has every reason to avoid contention and unhappiness among the rest of the partners.

Commitments of time

Partners should join the venture based on their ability to participate in it. We are not looking for investor partners who will not be involved in the operation of the company and its projects (although if one should stumble in, we'd be glad to talk). The principals of MSL are devoting their full time to this venture, limited only by ongoing commitments to MSL customers and prior consulting arrangements. Potential partners must decide for themselves how much time they have to devote to MSP. The basic quantity you should try to calculate is hours per week. We need an ongoing, reliable commitment of time by all participants. Whether you work two hours per day or in one fourteen-hour mad gonzo session each Saturday does not matter. If you have a job, however, which may randomly require your full absorption for a week at a time and leave you with stretches of idle time at random, that employment is not compatible with MSP partnership. We must be able to quote schedules and meet them, and we must be able to coordinate work from several implementors into final products. I know from experience that this cannot be done unless reliable time commitments are made.

The basic time commitments that participation in MSP entails boil down to the following three categories. First, the basic time devoted to company work which can be scheduled as you see fit. The time you have available for this work is the factor that determines the extent of your participation in the company. Second, each partner should be available for telephone conversation at some time during business hours on a daily basis. This is required for coordination of projects, passing on bug reports, or response to customer questions. If you can be reached at work, say, in the afternoons, that's all that's required. At an absolute crisis maximum, this would represent 15 minutes per day. Normally, one call per week would suffice. This refers to calls between headquarters and partners only, of course. If you're collaborating with another partner on a project, that time would be counted in the first category. Third, each partner should budget the time to attend partnership review meetings. These meetings will initially be held monthly on a regular schedule so that you can plan around them. We will alternate meetings among the various geographic areas where partners reside. If MSP includes partners not in the San Francisco area, we will make cassette tapes of the meetings available to those partners and accept written project summaries from them. This is not an attractive option, and remote partners should plan to increase the time for telephone consultation as a result.

Remember, this industry is now at a point where virtually all our competitors are ongoing operations with full-time technical employees. We're going up against them with less capital, a distributed operation, and less personal and financial commitment from the majority of our participants. We may very well fail. If we succeed

(and I wouldn't be getting into this unless the odds looked good to me) it will be because we know more about what we're doing than most of them technically (this I know for sure), because a partner always out-produces an employee, and because we have and will develop the contacts to aid us in product definition and marketing. But we're going to have to think lean and hungry for quite a while and target our products with precision. And most of all, we have to look, we have to *be*, a serious business venture, which we only marginally are. Most of the people who've succeeded in this game are those who sold their houses, quit their jobs, borrowed every penny they could scrape up, hired 5 or 10 people and hung their balls out over the abyss hoping their product would make it and bail them out. Making it while risking less is very very hard. I do not want to minimize this, but I want to point out that the risks in getting involved in MSP are probably less than any other serious business opportunity you're likely to find with anything like the potential return if it works.

I think we have a chance of making it with less than full-time commitments from partners only if their time commitments are utterly reliable. We're going to have to try to turn multiple part time people into the illusion of a full time staff so we can react to the market and bring out products as good as our competition and faster. That ain't easy. The people we're contacting as potential partners are the best computer people I know of in this country today, and are far better in both knowledge and productivity than the staff of most microcomputer software houses. That is what makes this possible at all.

The nature of potential products

I view the products that MSP will develop as falling into several distinct classes:

The first I call "guerrilla programming". This consists of developing relatively small, quickly implemented products which fill an immediate need perceived by users of a heavily promoted product. For example, a 3270-type screen oriented data entry package which generates SELECTOR files would be such a product. Every existing SELECTOR customer would be a prospect for our package, and systems houses who implemented applications in SELECTOR would use our package and sell it for us to their customers. A systems programming example of guerrilla programming would be a super-reliable file recovery program for CP/M. Again, every CP/M user would be a prospect for this utility. These kinds of programs tend to be quickly developed, sell fast, but don't last long as often the vendor you're tagging along with brings out a new release with your feature in it. However, they do make money and you can afford to do a lot of them since they don't take long to write. You can hit it big with one of these if, say, the vendor picks up your package and starts promoting it. This is not likely, and no project should count on this.

The second is the closed system application. This is a stand-alone application package which performs a well defined function for a specific class of users. Visi-Calc is a superb example of such an application. If you hit on one that's widely needed and not currently in a tolerable form on a micro you can do very well with these. Market research is essential here, and looking at what people are paying to do on timesharing systems is a good place to start. The "card file" very simple database is something we might do in this arena.

The third is the software tool. This is a utility program which is applicable to a wide variety of users for different purposes. Examples are SELECTOR and other database systems, word processing programs, and sort packages. This is a highly competitive market where large advertising budgets predominate and thus hard to break into. However, the rewards are great. We should look at somewhat "kinky" tools that haven't penetrated the micro market far but which have been popular on other systems. SSG and a SCCS-type facility are two that pop into my mind.

Fourth is the "interface gadget". We all do this well and they sell very well in the micro market. For example, a 3780 emulator, a CP/M to IBM disc convert utility, and so on. The problem is not being hardware dependent, and that's difficult in this game.

These categories overlap to some extent, but I think you get the drift of the kinds of things I'm thinking about. A good rule of thumb is that anything we do should fill a need the potential customer already knows he has, or should be demonstrated to a prospect in 5 minutes or less. We don't have the resources to educate the user base or to change the world. Products for which we can prepare a "demo disc" for computer stores are particularly attractive. We can give away a demo disc, then when a prospect walks into a store, they can run the disc which sells the package.

Hardware and system strategy

At this moment, the best established machine base for programs is the CP/M marketplace. There are about 500,000 machines installed which can run CP/M programs in one form or another, and the importance of this marketplace is underlined by the fact that most serious applications for the Apple now require the "softcard" with on-board Z-80 and CP/M.

However, the industry is changing rapidly and at this instant it appears that Unix or one of its look-alikes may become the "software bus" on 16 bit processors. We can't afford to bet on one system to the exclusion of all others. Fortunately, most of the potential products we're able to undertake don't require us to make a bet. We will be doing all of our programming in high level languages, and we must choose languages available on all of our potential target machines. At this date, C and Pascal meet this requirement.

We should seriously evaluate the option of going with CBASIC as our standard language and developing QBASIC implementations on the newer machines. The advantage of CBASIC (CB80 compiler) is that our work is file-compatible with a very large set of existing applications on CP/M, and with the acquisition of CBASIC by Digital Research (CP/M's developer), the connection is likely to strengthen.

On the other hand, the Microsoft-Unix/Xenix-IBM connection is a potent one, not to be ignored. I don't think we should be too bogged down by all of this, though. Whatever we program something in is going to generate object code that we distribute, and we're only going to program things which can be sold to a large number of customers without modification. If we do things reasonably, we'll be able to convert them to anything else that comes along and looks attractive. After all, conversions aren't fun, but if by converting something from CBASIC to Microsoft BASIC I can sell another 100,000 copies, I'll convert it. We shouldn't spend our time trying to figure out how many *SIGPLAN Notices* can dance on the head of a bit when we could be defining products, implementing them, selling them, and getting rich.

Why get involved?

If the tone of this paper so far has been to scare you away from this venture and to repetitively drum all the potential risks involved in joining such an operation, that's exactly what I intended. I've tried to lay out the whole operation complete with all the potential problems as straight as I can. So why would anybody in his right mind get involved in such a nutty venture?

The reason is very simple: there's a reasonable chance of making money beyond the wildest dream of an employee in this industry. Products like Wordstar are selling in the $10–20 million per year range today. Bear in mind—this is a product that any of us could write in about two months. We should consider ourselves extremely lucky to be in this business at this time in history. It's a rare piece of luck to have the field you've chosen as your career explode into the hottest growing entrepreneurial arena just as you hit your prime, and we're now at the point that if we want a chance to get involved we have to act immediately. The game has changed and the pace is accelerating very rapidly. The venture capital that remade the micro hardware business 2 years ago is just now beginning to move into the software business: within the last 3 months, Digital Research, Microsoft, Micro-Pro, and Lifeboat have received infusions of venture capital in the $1–10 million range. This business is getting very big and very professional, and within one year the chances of success of a tiny, heavily technically oriented company will be nil. If we move now, if we move fast, and if we react extremely rapidly and work ourselves to the bone, we can grab a chunk of this business before it slips away. We have to pursue our contacts at Lifeboat because that's an open door far too priceless to ignore, and we have to have a credible organization to open that door to further work.

If we sit back and say, "Well, I'll see how well the IBM makes out", or "Maybe after I pay off my car", or whatever, we'll lose a chance that won't come by again in our lifetimes. I think that with what we've learned from Marinchip and from the industry, that with the marketing contacts we have, with the product ideas we're kicking around, and with the competence of the people we know, we have a real enough chance to make it that it's worth betting everything on. But we have to have *real* commitment, *real* performance, *real* responsibility, and *real* professionalism to make it. If you're interested in making that kind of commitment, I can't guarantee that we'll succeed, but I can guarantee that together we'll have a once in a lifetime experience as we try.

What it all comes down to is the following questions, which only you can answer for yourself.

Do I really want to be in business for myself?
Do I want to work with these people?
Will I enjoy it if I participate in this?
Am I likely to find a better opportunity elsewhere?
Am I likely to find a better opportunity later on?
Can I manage the risk, and does the potential reward justify it?

I think that this is it.

Nitty-gritty

I have not discussed any of the specific details of the venture in this paper, such as the amount of money to capitalize the company, how much each limited partner would be expected to chip in, etc. Nor have I gone into specifics about the precise organization of the company or who does what. This just isn't possible yet; I have no idea who is really interested in it. You build an organization out of the people you have, you don't try to ram people into predefined slots.

We want to start a venture which in three years will be one of the top five names in the microcomputer software business. We're crazy to aim lower or limit our sights. We're at a point where substantial market segments haven't been addressed yet and by moving fast we can grab a market share and make our company grow from

generated revenues (note that all the software houses who've brought in venture capital had basically saturated their initial market first). At the point where we have to make that decision, we can be consoled by the fact that we'll already be millionaires.

I can think of no business (well, legal business) where we can start-up with so little capital or downside risk. If this business looks too shaky to you, where do you expect to find a better deal? I cannot imagine any scenario other than total collapse of society in which the sales of microcomputer application software will not grow by a factor of 10 in the next five years. The big vendors of small machines have not only not entered the software business, they appear totally in awe of it and willing to grab any product and promote it to sell their machines.

What do we do next

The first thing to do is to show up at our organization meeting at MSL on January 30, 1982. You should give some thought to the points raised in this paper about commitment of time, and should also be able to give an idea of how much money you'd be willing to risk on this venture (whether you have it or not). Also, we'd like an idea of what kinds of work you'd like to concentrate on, and any ideas you have for products we might get into. In particular, if there are any items in this paper that are "show stoppers" for you or with which you take violent exception, that's the time to bring 'em up and hammer them out. At the end of that meeting, which will probably be very long and detailed, I hope that those who are interested in proceeding know who they are. Then we'll start putting numbers on paper and see what we're getting into.

We should shoot for having the company in operation by mid-March. We cannot dawdle, but we also are going to do it right this time. We're just going to do it fast!

Agenda

Agenda for January 30, 1982 Meeting

- Overview of MSP
  - Goals
  - Marketing targets
  - Potential product areas
  - Marketing channels
- Background
  - The micro industry today
    - mass market hardware
    - software development
    - software marketing
  - Marinchip—what we've learned
  - Case studies
    - Micro-pro
    - Scripsit
- What's needed to succeed
  - Market-directed products
  - "Don't be afraid not to innovate"
  - Responsive organization
  - Marketing follow-up and project monitoring
  - Highest standard of products from first release
  - Target expanding mass markets
  - Sufficient capital and commitment
  - Afford to be wrong 80% of the time
- The difference between strategy and prediction
  - Make any potential success a success
  - Resources to keep on trying until you hit
  - Structure so you know when something's hitting
  - Organization which can swing behind a success
  - Ability to pyramid success when it occurs
- What MSP participation gives you
  - Full-time support operation

  - Marketing contacts
  - Market research contacts
  - Complete manufacturing operation
  - Risk capital for start-up

- What MSP wants from you
  - Commitment to MSP as best prospect to get rich
  - Meeting all delivery and support commitments
  - Providing marketing support as required
  - Production of highest quality products
  - Sharing ideas and information with others
  - Aiding others with partnership projects
  - Exclusive access to your work in areas MSP addresses
  - Your capital contribution
  - A level of effort you can maintain

- Don't expect MSP to...
  - Produce technological breakthroughs
  - Do pure research
  - Be as much fun as hacking
  - Spare you anxiety
  - Let you specialize
  - Exploit your existing knowledge optimally
  - Fit perfectly with your current job

- Expect MSP to...
  - Broaden your horizons beyond your imagination
  - Educate you in the realities of business
  - Teach you marketing
  - Make you appreciate the value in ideas you may disdain
  - Expose you to many different systems
  - Introduce you to depths of despair and exhaustion you never knew existed
  - Introduce you to heights of exultation you never knew existed
  - Ruin you for being an employee
  - Make you rich

- The last train out
  - Entry of venture capital to software business
  - Analogy with hardware business in '79—80

  - Difficulty of start-up venture in high-stakes game
  - The tension—demand for software vs. supply / difficulty to produce software in large organizations
  - Realities of introducing and promoting a product
  - Why we have a chance at all
- The open track ahead
  - Massive promotion of small machines in business environments
  - IBM sales staff consolidation
  - Dearth of software for desktop applications
  - Availability of growth capital / cash out opportunities
- Our experience and goals
  - Why the low-commitment game is over
  - Grow or die—Shrayer vs. Micro-Pro
  - Cash is needed up-front
  - Marketing follow-up and project evaluation is essential
  - Go for it—now is the time the GM's of the 2020's are being formed
  - What do they have we don't?
- Why get involved?
  - Can always think of something better, are you likely to find it?
  - Absolutely unique opportunity
  - Every incentive toward being in business
  - Cannot make it on your own
  - Why trust these turkeys? — I do $60K worth.
- Marinchip's contribution
  - Marinchip annual report
  - Liquid asset summary
  - Initial capitalization proposal
  - Marinchip ongoing operation facilities
- Partner contributions - round table
  - What partner has to contribute
    - business experience
    - technical experience
    - risk capital
  - What share partner is interested in
  - What skills partner wants to acquire
  - Any limitations on partner's participation?

    - maximum time commitment
    - won't quit my job regardless
    - won't work on .... (databases, sorts, compilers)
    - won't get involved in ...(marketing, ad copy, documentation)
    - won't work with others (or specific people)
  - What are your worries?
  - What have we left out?

- What I think is needed
  - Try to succeed, not prove something
  - Don't assume that because it's been done it's been done for all time
  - Distinguish your product from the rest, but don't make it so different it's incomprehensible.
  - The human mind's basic primitive is "this is like that, except...", learn to live with that.
  - Don't try to solve all problems for all time.
  - Don't offer any options, ever!
  - No configuration, ever!
  - No system programmer after you.

- Why I think we can do it
  - We have the technical competence edge on almost everybody
  - We're building a responsive structure, and *we will make it work!*
  - We have the slant and contacts—micros are moving from the beachhead into the mainframe application class.
  - We know mainframes and what people do with them and how.
  - We've always been able to beat anybody on delivery time if we really care to.
  - We have the systems programming capability to back up our applications. None of our competitors do.
  - We have a comfortable amount of seed capital—no need to bootstrap or to produce instant performance for outside funding.
  - We have the historical perspective—almost none of our competitors has been in computer for more than 5 years.
  - We're able to adapt ourselves to the market—we're not gambling everything on one product or concept.

- Timetable

  2/6 Participation commitments due.

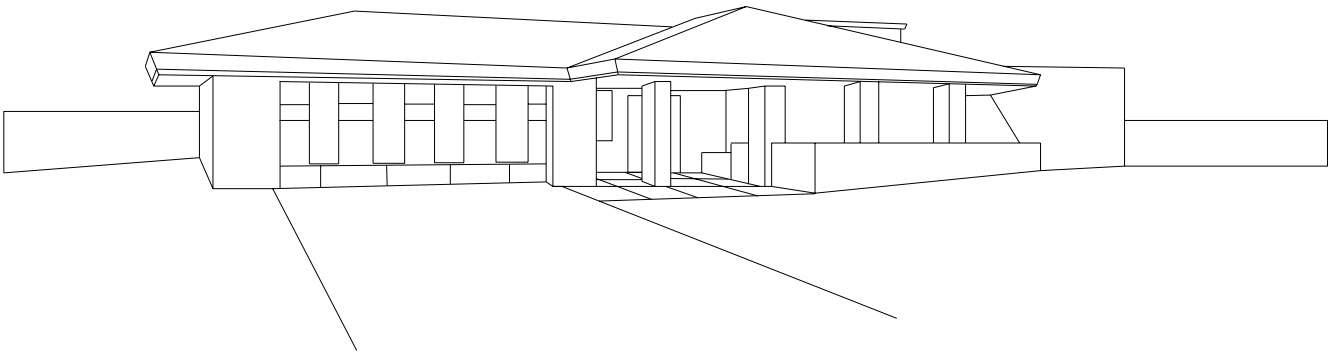
  2/8-13 Partnership share consulations, draft agreement review.

  2/13 Partnership charter meeting—tentative agreement approval, projects review, initial work assignments, hardware procurement review.

  2/27 Formal partnership organization—agreements signed, initial capitalization delivered.

  3/13 First partnership review meeting.

This drawing, originally done on M9900 INTERACT, was the first architectural drawing ever used with AutoCAD. It was shown at the introduction of AutoCAD at COMDEX in 1982.



(C) 1978 Frank Lloyd Wright Foundation
used by permission

Information Letter 1

After the original working paper was mailed, I had numerous telephone conversations with the people who received it. I wrote the first Information Letter and mailed it before the meeting in an attempt to address several points which came up repeatedly in these conversations. This letter inaugurated the series of Information Letters which have continued up to the present day.

Marin Software Partners
Information Letter # 1

by John Walker
Revision 3 — January 19, 1982

This letter is intended to clarify some points in the Marin Software Partners Working Paper which you should have already received, and also to bring you up to date on some discussions about matters not covered in the original paper. The organization meeting will be at 11:00 A.M. January 30, 1982 at Marinchip.

Structure of the business

Most of the questions recipients of the Working Paper have had related to the partnership structure of Marin Software Partners. I'll try to clarify what we want, point out potential problems, and look at alternatives.

Let's start with basics. We are planning to organize a business in which all the founders will have an "equity stake"—in plain language they, collectively, will be the owners of the business. There are three basic ways such a business can be structured. The first, and most simple, is the general partnership. This is what is usually meant when the word "partnership" is used. In a general partnership, all partners participate in the operation of the business and share in its profits or losses proportional to their ownership share. Each general partner is subject to unlimited liability in regard to the operation of the business, and is liable for debts incurred in the partnership's name by any of the general partners. This means that a general partner who lacks a controlling share of the business must trust the other partners sufficiently to expose himself to unlimited losses if they misjudge or act improperly. Steps can be taken to hedge this risk, such as an insurance policy against liability suits, embezzlement, etc., and statements in the partnership agreement which allow borrowing only if approved by unanimous consent of partners.

A limited partnership is composed of one or more general partners, who have the same responsibilities and liabilities as before, and one or more limited partners whose liability is limited to their initial investment. Thus, while a limited partner may lose everything he contributed to the partnership, he may not lose more. In this sense, a limited partner is like a stockholder in a corporation. Limited partners do not have any direct say in the operation of the company—this is the function of the general partner(s). The most common application of the limited partnership is a venture where people want to put up money for a venture in the hope of sharing in its profits. The general partner takes the money, does something with it, and distributes the appropriate share of the proceeds back to the limited partners. This is a common way of financing oil and gas exploration, and is used extensively in the venture capital business. In any case, the limited partner retains an unlimited upside potential gain while limiting his loss to his original investment.

The third way of organizing a business is incorporating it. A corporation is by far the most flexible form of organization, but it has several important drawbacks for a venture of this kind. Most critical is the question of royalty income. The IRS holds that a corporation with 15 or fewer stockholders which receives a majority of its income from "passive sources" such as royalties and interest is a "personal holding company" and is subject to a 70% tax rate. This ruling was introduced to prevent people from forming companies to hold their investments and by so doing paying far less tax on the income than if they were taxed at the higher marginal rate which would result if the income were added to their regular employment income. The problem is that a software company may very well receive all its income from royalties if it licenses vendors and distributors to manufacture and sell its products, and this leaves such firms potentially liable under this provision. We have asked an attorney knowledgeable in this area about this problem, and he says that to date no software firm has been penalized under this ruling, but that there is no question that they are potentially liable. Nobody has even asked for an IRS ruling out of fear of alerting them to this source of income. The upshot of this is that if you intend to sell software for royalties, you'd better not incorporate until you're big enough to escape the provision through the size test.

The second drawback of a corporation is that income from the corporation is taxed twice: first as "corporate income", then again when it is received by shareholders as dividend income. Federal law allows you to declare a corporation "Subchapter S", which means that income is distributed directly to holders and no corporate tax is charged. This neatly solves the problem (unless you're too big for this treatment), except that good ol' California doesn't recognise Subchapter S. This means that you still pay double tax to California on your income. With California marginal tax rates at 11%, this is substantial dollars. Thus, most California Sub S corporations list the principals of the company as employees and try to pay out all their profits as salaries to the principals. Since salaries are deductible from corporate income, this avoids the problem. But there are some catches. First, the tax people have every incentive to bust you if it looks too much like you're using the salaries as a sham for dividends (which is exactly what you're doing, of course) since they make less that way. Second, you have to pay Social Security tax, unemployment tax, carpet tax, etc., on salary payments. Thus, there's really no clean solution to this problem. You say, "well, why not just leave the money in the company and let it collect interest at the low corporate tax rate until I need it". Gotcha again! You can't leave more than $150,000 in the company unless you can show it's legitimately needed without triggering punitive measures.

Third, a corporation is a pain in the ass to run. The workload involved in filling out forms, filing statements with the state and IRS, etc., is easily 5 times that of a partnership of similar size. It takes about a tenth of a full time person to do, and unless there's an obvious return, it's therefore to be avoided.

The benefits of the corporation are also often overstated. The much vaunted limited liability of the corporation can evaporate if the corporation can be shown to be a sham set up only to limit the liability. In any case, directors of corporations are subject to lawsuit against their personal assets based on the acts of the corporation. Thus, it doesn't look to me like there's any reason to consider incorporating this business at this time.

Since we already have a corporation, Marinchip Systems Ltd., we can make that corporation a general partner in Marin Software Partners. It may be possible to structure things so that the liability of the partners is limited, and the principals of MSL are protected to the extent that the corporation provides. We will have to seek legal advice as to whether this will work and how it must be done, and we will do this as we form the new venture. It just looks like this is the best option if it will work.

The issue of liability

I don't want to get everybody all worked up about this issue of liability. In deciding how to organize a new business, you have to balance various considerations. You want a structure in which the responsibility is assigned in proportion to the commitment of the various participants. You want to limit the potential loss a participant can sustain as much as possible. You want to minimize the percentage of your gains that the tax man will pocket. You want an organization which can grow and change without catastrophic problems.

From almost every consideration except that of taxes, a corporation is the best. But since we have the potentially disastrous royalty tax problem, we can't incorporate safely. Thus, a discussion of liability is in order as that is the major difference in the two forms of partnership. This business is very different from most start-up ventures. We are capitalizing the company by contributing money to the original "pot", and are creating our products almost entirely out of mental effort. We have no employees, and our fixed costs are almost negligible. We will have no debts to anybody (I am a fanatic about this), and our capital investment will be a relatively small percentage of net worth. The software business is just about optimal from the standpoint of product liability. Programs blow up in people's faces only figuratively, and unlike most other things, software is usually sold with no warranty, or one limited to refund of the purchase price within, say, 30 days.

As a result, the liability exposure of a partner in this venture is about as small as could be possible in any business. Basically, if we reach the point where the bank account is drawn down to zero and we haven't sold anything, we fold up our tent and go back to the salt mines. It's hard to imagine how you could lose more than your investment in this kind of venture. Thus, being a general partner is not the risky thing it is in a real estate venture where you're signing up to be liable for a chunk of a $30 million floating rate construction loan on a building for which there may be no demand when it's finished 24 months from now. I don't think the liability issue here is something to lose sleep over, and I personally don't care if I'm a limited or general partner. I do realize that from your questions some of you are concerned about it, so I've tried to beat the issue to death looking at it from all sides here. That's not to say that any issue in forming a company is unimportant, but I think that you have to look at it in the light of the nature of the business, and in this case liability is not an overwhelming problem.

Changes after the business is running

Most of your questions have been related to how the company may change after it has been in operation for some time. I'd like to discuss some of these issues here. The problems you've raised are problems that I know no complete, clean solutions to. However, business, like life, is an endless series of problems to overcome. You always try to avoid problems where possible and mitigate the effects of the ones you run into. But if you refuse to do something because there are potential problems, you end up never doing anything. But enough preaching... on to the grimy details.

"How do we handle it if one partner doesn't meet his commitments?" This has been the most commonly asked question, indicating that I am no more cynical about participants in business ventures than you. Obviously, we only want to go into business with people we respect and trust, and the first level of screening is performed when we get together and decide who wants to work together in this venture. If somebody seems not to fit, either based on their goals, their approach to the venture, or their ability to get along with and work with the others, it would be a mistake for all concerned for that person to become involved in Marin Software Partners. Of course, you can make a mistake. I made such a mistake in my selection of an original partner in Marinchip, so I'm very aware of this possibility. The solution in that case was what usually proves best; the rest of the partners buy out the partner who has lost interest in the venture. If the business is prospering, this buy-out can be paid for out of sales revenues. If the business is failing, the partner's share is not likely to be worth much in the first place. I know of no solution to the case of the partner who just refuses to work or becomes obstructive but who refuses to be bought out. So we don't include any assholes, O.K.? (It might be possible to write the agreement in such a way that the other partners retain an option to buy out a partner for a certain price for a stated term. I think this is a terrible idea, since it would put the financially strong partners in a position where, if the business "took off", they could grab the business from the less strong. While, of course, nobody involved in this venture would think of such a thing, it wouldn't contribute to a partner's equanimity knowing such a coup were possible.)

"Are decisions affecting the partnership made by unanimous consent or by majority of partnership share?" This question is really irrelevant if the limited partnership is used, as MSL would make all decisions. I also don't know the relevant law (we will, of course, find out as we work with a lawyer to draft the partnership agreement, although I'd like to believe you can have anything you want put in the agreement). I think that the majority share makes the most sense, even though it has its obvious risks. After all, all stock corporations work that way and they seem to make out all right. I'd be worried about one stubborn person being able to immobilize the entire company (after all, I've been known to be stubborn—and dead wrong—myself).

"Suppose the company takes off and I want to quit my job and do this full time. How can I increase my share?" In this case, you would purchase an additional share in the partnership at a price agreed to by the other partners. Your share might be sold to you by another partner who wished to reduce his share (MSL might want to "cash out" to free up money for other ventures, for example), or could be a new share which effectively dilutes the shares of all the other partners. The partners whose shares were being diluted by this act would be compensated by the payment you made for your additional share, and by the presumed increase in revenues which would result from the additional work you did for the company. The price you pay for your additional share is the price the other partners agree to sell it to you for. If the company is a corporation, change "partnership share" to "shares of stock" and everything is the same.

"How do we bring in new people?" In the case of new partners, the case is exactly the same as that discussed above for an existing partner increasing his share. Of course, a new partner may buy in by supplying any form of consideration, such as rights to a software package he had developed. If we decide to expand the general operation side of the business, we may decide to add some conventional employees. This would just involve salaries paid out of the general revenues of the business and doesn't affect the partnership in any way.

"Suppose I want out?" This is just the reverse of the case of adding to your share. You sell your share back the other partners for whatever they're willing to pay you for it. They recover what they paid you by the increased shares they own after yours are liquidated. Of course, if they don't want to sell, you're stuck. They'd be stupid not to, though, for otherwise they would have to continue to pay you your partnership share of the revenues in return for your doing no work.

"I'm worried about having the business expand to the point where I have to quit my job. What do I do if this

happens?" This is the kind of problem that is good to have. Basically, you have to calculate the equity value of your job, which is just like valuing a company: what is the income, how secure is it, what are the growth prospects, and what is the equity I sacrifice by quitting (seniority, pension fund equity, future employment prospects, etc.). If the job is so valuable to you you'd never quit, then you shouldn't consider going into business for yourself. If the job has a value (they all really do, of course), then you should only quit to take a job with greater value. I'd quit a $50K job with, say, Consolidated Engine Sludge to take a $10K job with Advanced Robotic Widgets if I had a stock option for 20% of the company, but everybody has a different situation and has to make his own decision. Working out the value of your job in your head is a worthwhile effort in any case as it gives you a better perspective on what you've got. I believe everybody constantly acts to maximize their overall gain in life (not just economic, of course; personal happiness, adventure, are calculated in as well), and that if the time comes where participation in Marin Software Partners is seen as better than your current job, you'll have no trouble making the plunge. Those that can't are the Hamlet types who never do anything and always fail in business anyway.

"I'm not sure I have the experience to be in a company like this." I have found that a sincere desire to be in business for oneself and to work hard is far more important to success in business than detailed technical knowledge. If you can do a job well and the partnership needs that job done, you belong in Marin Software Partners. We'll have to look at the mix of talents we have in the people who are interested in forming Marin Software Partners and address the areas where we're deficient. We'll almost probably have to go outside for advertising preparation, but we'll probably have the in-house capacity for all the technical writing we need. At this point we can't say whether *anybody* fits into the company—we can evaluate that only after we see who's interested and who's not. Being in business is an excellent way to learn about thousand of things you never intended to learn about. If you're looking to learn new things and expand your horizons, this is one way to do it (although getting run over by a truck may be less painful).

Terminating products

Some people have expressed concern about the continuing support burden of products we decide to terminate because of bad sales. This is a non-issue. When a product is terminated, support of it is terminated and those who inquire are simply told "that product is discontinued". Only in the computer business does the insane idea that by buying a $300 product (or, for heaven's sake, a $35 product) entitle the purchaser to all products of the implementor's mind unto eternity and unlimited free consultation at the press of a touchtone button. If you buy a refrigerator, you don't expect to get a new one every 6 months because a new model comes out, and the same thing holds here. We warrant the product will agree with the manual and will work for, say, 6 months after purchase. When that expires, our connection with that product is totally severed. We may choose to offer existing customers a good deal on new versions, but that is a marketing tactic, *not a moral imperative*! If we find the mass market we seek, we can't afford to talk to one in a hundred end users. We have to make the software work in such a way, supported by the manuals, that users can use the package on their own, and provide the aids to those who distribute the software so that they can answer user questions locally. This isn't impossible: numerous products meeting this criterion sell well currently. So, when we terminate a product, it's done.

What to call the company

I'm using Marin Software Partners as a working name for the company. I haven't thought at all about what it will really be called and don't think this is of any importance at this time.

How much of a share to buy

Several people have asked about how much of a share of the company they would be expected to buy and how much it would cost. It's hard to put numbers on this until we see who's interested and at what level of effort, but I can give you some rough background on how this will eventually be crunched out.

The first question any partner must ask, as detailed in the Working Paper, is "How much time can I commit to the venture". Once you know that (and I'm assuming everybody will be able to say at the organization meeting), we can begin to guess at a share. Ideally, everybody should buy in at a share proportional to their time commitment. Let's say that of the technical partners, a total of 100 hours per week is available. Let's assume than MSL is buying 60% of the company by its cash contribution and the technical partners are buying a total of 40%, and that MSL is contributing $50,000 to found the company.

Now if you have 15 hours per week to work for Marin Software Partners, your ideal share would be 15/100 or 15% of the 40% owned by the technical partners over all. Thus, your share of the total company would be 6%. To purchase this share for cash, your contribution would be $(50000/0.60) \times 0.06$, or $5,000. If all the technical partners purchased their shares of the company for cash, the company would start with an initial capitalization of $83,333. If the company made $80,000 profits in the second year (this is the level MSL was running at its peak), then the partner's share would be paid off at this point. If the company reaches $1,000,000 in profits then the partner's original $5,000 investment will be yielding $60,000 per year in income.

If you don't have the $5,000, but want to buy in at that level, we'll find the money for you to borrow. If that number just looks too doggone big, you might choose to buy a lesser share and reduce your time share accordingly. This is a gamble, of course, since if the business begins to grow rapidly and you wish to increase your share, it will cost you far more to do so (as a percent of a company earning $100,000 per year is worth a lot more than one of a company earning zero). Conversely, if the business starts off poorly, you might be able to pick up a piece for less.

Order of magnitude financial figures

Again, it's too early to put down hard and fast numbers, but you've asked, and need to know just what the magnitude of this venture is going to be. My gut feeling is that the absolute bare bones Spartan scrape by high risk minimum to get started in this business is $50,000 initial capital with as little of that spent on hardware and fixtures as possible. I'd feel comfortable with $100,000, and feel that if we couldn't make it with that, then we probably couldn't make it with a million. MSL has about $50,000 in liquid assets to funnel into this venture, so I think we can plan to scrape up the kind of money to give us a reasonable shot at success. MSL's fiscal year closes on January 31, so on the 30'th we'll pass out our preliminary financial statement and let you see where MSL stands in this venture.

Once we see just what we've got, we may have to line up some additional financing. One possibility is for somebody, like me, who's got some cash and believes in this venture to grant Marin Software Partners a line of credit which it could draw on, at, say 5% above the prime. This would provide Marin Software Partners with a source of additional funds if it needed them, which it would really have to to pay my usurious (but utterly risk-justified) rates.

The role of MSL principals

We've been asked about what role the principals in MSL (John Walker and Dan Drake) will play in the partnership. They will participate in the operation of the company and share its results by virtue of their ownership and operation of MSL. In addition, they may choose, as individuals, to become limited partners in the venture. They would do so based on their desire to work, exactly like other limited partners, on technical projects to be marketed by Marin Software Partners. If they do so (and I certainly haven't made up my mind—I *want* to, but where is the time to come from?), they will buy their shares just like anybody else and participate on the same basis.

This is an issue that isn't settled yet, and I don't think it's very important until we see just who's interested.

The promise of this venture

Everything you've read so far relating to this company has stressed the risks of any business venture, the depth of the commitment involved, and the potential problems and catastrophes which can befall one who dares venture from the corporate womb. I keep hammering on these points because I've found that underestimating them is the most common problem people have when going into business. I don't want you to conclude, however, that I lack enthusiasm for this venture or that I expect it to fail.

I think that we're at an absolutely unprecedented juncture of history. I can't think of any time in the entire human experience when so much opportunity existed for technical people, opportunity which they could participate in with very limited risk. Most great business opportunities have required far greater infusions of start up capital which was consumed just paying for physical plant before anything was made to sell. Our products are created by almost pure mental effort, and are manufactured on trivially cheap equipment at a tiny fraction of their wholesale cost. It's almost like counterfeiting, but legal.[7]

At the same time, we're entering a marketplace which is expanding at an unbelievable rate. Wander through any office tower in downtown San Francisco and look at how many desks have computers on them. Say, less than 1%. In 5 years or so, 80 to 100% of those desks are going to have computers on them, and those computers will be running programs that have not been written yet. In less than 6 weeks, over 100,000 IBM personal computers have been sold. There is little or no serious application software for that machine at present. If we make $100 per copy on a database system for that machine, and sell 50% of those customers on it, we've pocketed five million bucks. And how many will they sell in the next five years....

The potential rewards of this business, which is the field that you and I are technically proficient in, almost compel one to participate on an equity basis. There's almost no salary that's enough to reward one for giving

[7] I later came to refer to the software business as "100% value-added—pure reason without the critique".

up his seat at this cosmic money gusher. There's no doubt that we have the technical proficiency to produce products as good as those any of our competitors are selling. The quality of our technical writing is continually mentioned as being superb. We know we are incompetent at advertising, but we know where to purchase that talent at a reasonable price (at least, compared to page rates). We should have at least one salesman-partner—this is a serious lack and if you have any contacts in mind please forward them. If we have to do without, though, we're not doomed, as we're already plugged into the marketing channels through Lifeboat, who already has a sales organization targeted at our market. There are enough precedents for the "strong technical, weak sales" company making it on the strength of their products to convince me that this business isn't like selling toothpaste.

The last thing I want to do is to sell anybody on getting involved in this venture who is less enthusiastic about it than I am. But I can't help saying to anybody who doesn't want to get involved, "Do you ever expect to see an opportunity this good come around again in the rest of your life?". I'm not talking about this company specifically, as you might have legitimate worries about getting involved with the people and slant of this particular venture; I'm saying that here we have an exploding market that you understand technically, which can be entered with little capital, where huge corporations are trying to promote your work to sell their hardware, where growth capital is readily available when it's needed, and where there's still time to get in without being an employee or minor stockholder in a big venture. 99.99% of all the people in this country live their lives without ever having the kind of opportunity we have in front of us today. Those who do not choose to take it should not count on it knocking again.

Responsiveness

Some people have interpreted the product development structure suggested in the Working Paper as smacking too much of the kind of bureaucracy they dislike in their current jobs. I think that one of the great advantages a small company has is its ability to react rapidly and get things done before the competition does, and that any hardening of the arteries which prevents this spells disaster. On the other hand, if the organization is so loose that you don't know where the money is coming from, how can you decide what you should be doing? My goal is to provide the minimum level of structure needed to define, develop, market, and promote products, targeted at all times to maximizing our profits. I think the structure I suggested meets this criterion. I can think of nothing less that will serve. On reflection, I think that it may be wise to designate one partner as "product manager" for each product, even if several partners are collaborating on it. This isn't to introduce unnecessary hierarchy, but simply to provide one person who will coordinate user communications, contact with headquarters, integration of work by other partners, etc.

Philosophy

To those of you who know the esteem in which I hold Don Lancaster's book, *The Incredible Secret Money Machine*,[8] some of the concepts you've seen here may seem alien or repugnant. My theme all along in this is "the game has changed". To be blunt, they're playing the ball game with real balls now. It's possible to follow the Don Lancaster route and earn a reasonable income for life while maintaining your own freedom and lifestyle, but you only generate income when you work, and you must resign yourself to seeing people with less merit in your eyes advance beyond you on the ladder of material success.

[8] *The Incredible Secret Money Machine* by Don Lancaster, Howard W. Sams, 1978. *Read this book!*

Everybody has to decide what's important to them. For some people, it's knowing that their work is the best. For others, it's understanding things. Some people measure their value by certificates on their walls, still others by certificates in their safe deposit boxes.

If you're a computer person, you have a rare skill that's much in demand. If you can produce a given kind of work, I think you should demand as much for it as you can get, consistent with the constraints you're willing to accept on your lifestyle. My own personal belief is that in a venture of this kind I have a reasonable prospect of realizing far more for my efforts than from anything else I could do with my time. I think that by taking a less serious approach to this business you only reduce your prospects of success. Thus, today, I think this is the best option, which is why I'm willing to bet so much on its success. If we fail, we'll lose sums of money which are significant, but which are in the nature of a setback, not a disaster. If we succeed, we'll be able to put the whole job game behind us. And then, hopefully, relax and enjoy the fine things life has to offer.

Potential Products for MSP

Existing products: The following products exist already under various ownership and are being marketed by MSL or to MSL customers. We assume that title to the products would transfer to MSP in return for payment by MSP to their owners or partnership share.

Existing products

INTERACT. INTERACT[9] could be supported by MSP on the IBM personal computer, or possibly on the Tandy or Apple 68000 machines to be announced shortly.[10] INTERACT is written in SPL and an SPL-port would be implied in any conversion to a new machine. Since the trend in machines is to better graphics and faster processors, each movement along this trend makes INTERACT a more attractive product.

The major drawback is that INTERACT needs either a hard disc or at the minimum DS/DD 8 inch floppies to run. This lets out most of the current desktop mass market machines unless somehow INTERACT's dynamic drawing file can be compressed. The major advantage is that INTERACT is a superb product in a virgin market.

QBASIC. By making QBASIC[11] CB80 compatible and porting it to the 8086 and 68000, we can establish a beachhead in the 16 bit system software market. We are basically counting on outrunning Gordon Eubanks[12] and sneaking in below Digital Research's advertising blitz. Probably the best strategy is to continue to pursue OEM buy-outs through Lifeboat, as they have a large incentive in reducing their royalty payments to D.R. And of course, QBASIC is now extremely easy to port and each new processor is a new product. Also, we can use QBASIC for our own applications work.

[9] INTERACT was the product that formed the starting point for the development of AutoCAD. This is the first suggestion of a CAD package as a potential product for the company.

[10] The "Apple 68000 machine" was the Lisa. The Macintosh came much later.

[11] QBASIC was a compiler for the Marinchip machine which was language-compatible with CBASIC, a popular business dialect of BASIC, then available on the 8086 only as an interpreter.

[12] The developer of CBASIC, then at Digital Research.

C. Dan Gochnauer's full C for the Z8000 is written in itself and can be ported to other machines. I don't know if this is of any real value except as a low-cost entry in an already congested market. Maybe we should slap the sucker on the IBM and promote it like we just invented C. This won't make us rich, but it may make us some money.

SORT. Hal Royaltey's SORT package should be reviewed in the light of competitive products under CP/M. If it shows potential, we can enter the SORT derby under CP/M with it. Maybe we can think of some sexy gewgaw to make it stand out from the pack. Ideas?

WINDOW. Probably no market. Damn shame.[13]

SPELL. Probably no market except at $25 a pop.

LENS. We might be able to clean this[14] up and put it in CBASIC (it has no assembly language in it) and sell it through *Sky & Telescope* to CP/M'ers. Probably some "feechers" should be added first, though.

DIFF. The current QBASIC version can be enhanced to do CP/M directories, put under CB80 and peddled to CP/M'ers.[15]

SELECTOR. We might be able to negotiate a distributorship for SELECTOR V on the IBM in return for putting it up in QBASIC. This might prove to be very lucrative even if our cut was very small.

New products

The following are product ideas of various degrees of definition which might fit into our new line of business.

Executive planning aid. This is being investigated by John Nagle. It's a screen oriented PERT package with costing and resource allocation, and every manager in a large company with a desktop computer is a potential prospect. We want to target products to this market segment as it is being aggressively targeted by IBM and Xerox and is likely to be one of the fastest growing market segments in the next few years.

[13] WINDOW was Marinchip's screen editor, itself written in QBASIC. I wrote WINDOW originally to squelch the outcry from Marinchip customers for a "screen editor"—something I believed at the time beneath real programmers. I named it WINDOW after the code name for radar chaff, a lightweight countermeasure that's all image and no substance. WINDOW was briefly marketed as AutoScreen for CP/M, and later its design served as the basis for the screen editing features of Kern Sibbald's editor.

[14] This was an interactive lens design program written in QBASIC for the Marinchip. Years later, translated to C, its central algorithm became the Autodesk floating point benchmark, cited on pages 324 and 560.

[15] Duff Kurland did this, but we never sold it. Later we converted it to C, and in that form still use it today.

Cardfile. This is being kicked around by John Walker. This is an ultra-simple database which lets you replace things you currently keep on scraps of paper or boxes of file cards. It requires absolutely no knowledge other than how to turn on the computer and type and works in a language as close to plain English as possible. I think that even if I had SELECTOR, I would still want a product like this.[16]

MAPPER. MAPPER is the first product Univac has developed which is being heavily promoted as a product in my memory. It is responsible for the sale of numerous very expensive systems simply to run it. We should study it carefully and see if it contains concepts which can be applied to a standalone desktop system. If so, is such a product applicable to any office, or is it salable only to the Univac user base? Might it fit as a product under UNIX?

Forms generator. This is a utility which allows people to design forms, and optionally fill them in. The simplest use is just to allow people to edit forms which are printed on a printer and used as Xerox masters. The stored form can also be used as input to a prompting routine which allows users to fill in the forms on the screen and generate either data files for input to other programs, or simple printed forms with the blanks filled in. This seems like a natural for the transitional "office of the future" which hasn't sworn off paper.

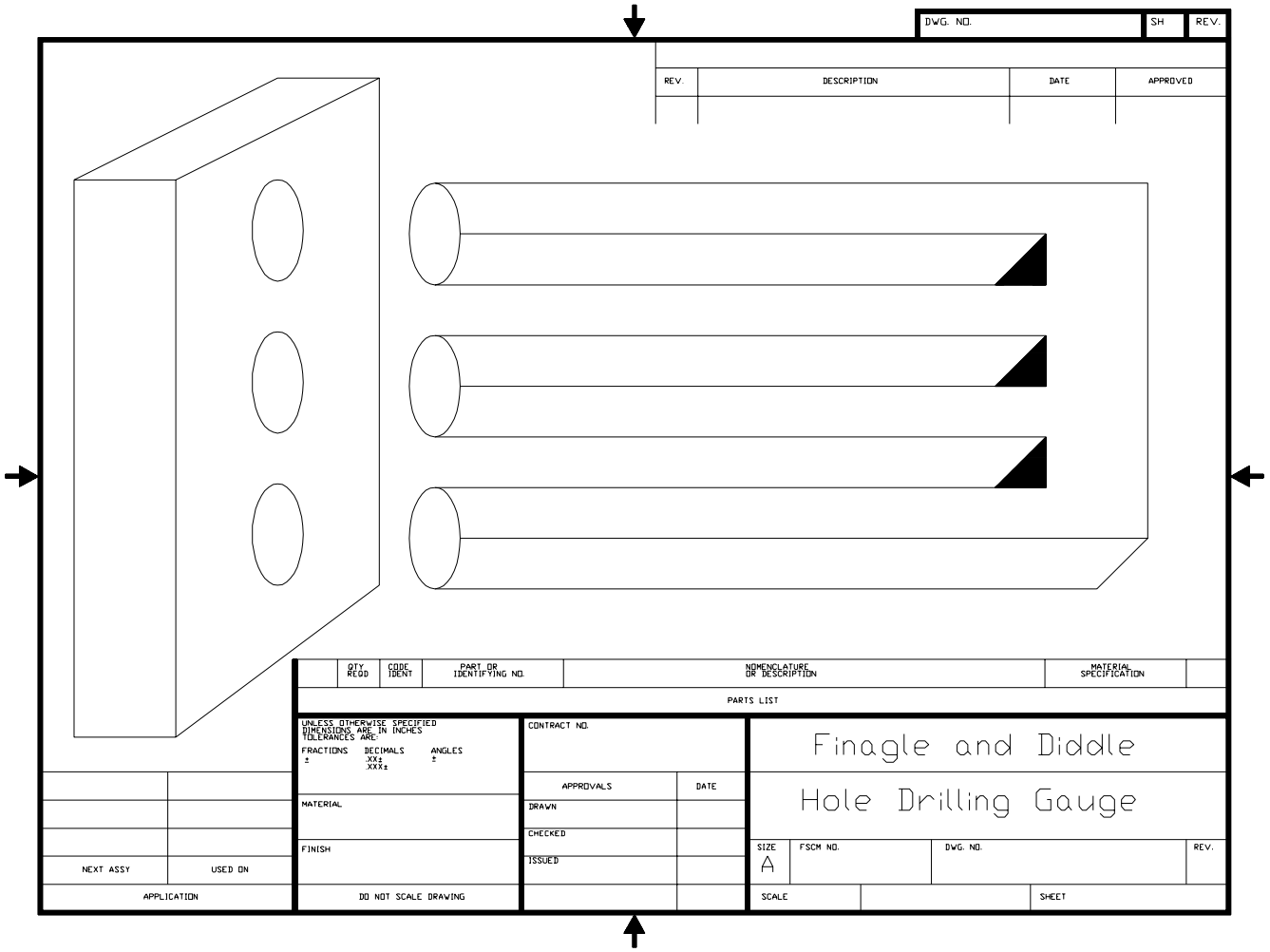
Menu-oriented TS. A terminal emulator which can be programmed to present menus and conduct dialogues with the remote system for the user.

JPLDIS. Convert it and sell on small systems.[17]

[16]This is the idea which grew into the product called Autodesk, after which the company was eventually named. It is amusing to compare the goals of this product, and of Autodesk, to the darling of 1988, HyperCard.

[17]We did not know, at this time, that dBASE II was a derivative of JPLDIS.

This drawing was done on AutoCAD-80 shortly before COMDEX 1982, and was shown at COMDEX as an example of a "mechanical drawing". I hand-measured an ANSI A size title block and drew the title block piece by piece. The ellipses were done by inserting circles with differential scale.



Information Letter 2

Information Letter 2 was the first general communication after the initial organisation meeting.

Information Letter # 2

by John Walker
Revision 1 — February 12, 1982

This letter is to bring you up to date on what has happened with the formation of MSP since the January 30, 1982 meeting. I assume that everybody who intends to participate has already sent a letter to that effect, so we know who is involved and what they want to do. Since just about everybody at the meeting decided they wanted in, there is certainly no doubt that our software development capability is awesome—I can think of no product on the microcomputer market today which we could not develop if we decided to. Now we have to put together the organization to define the products, produce them with the desired quality, and market them.

Alternate forms of organisation

Keith Marcelius suggested an alternate way of organising the company which looks to me like a potential solution to some of the major concerns we all had about the original proposal. It allows us to accommodate people whose financial contribution cannot be commensurate with their time to devote to the venture and it gives a way to reward those who contribute more than their expected share.

Let's assume for the moment that the company is formed as a corporation (this might also work for a limited partnership, but we don't know yet). Suppose we authorize and issue 1 million shares of stock initially (the number is totally irrelevant, but should be large enough so that round-off can be ignored). 600,000 shares of stock are sold to the founders of the company based on their capital contributions; this establishes their initial share. The number of shares purchased would be:

$$(YourContribution/TotalInitialCapital) \times 600000$$

The remaining 400,000 shares of stock would be held in the corporate treasury. The effect of these shares would be nil as long as they are retained in the treasury; if dividends are distributed they just loop out of the checking account back into the treasury.

Every year, based on people's contributions of work, a stock dividend can be declared to those stockholders who contributed in excess of their share. This means that we take those shares out of the treasury and give them to the person who contributed the extra time. This increases his share at the cost of diluting the shares of those who did not receive the stock dividends. All distribution of stock dividends would be subject to a majority vote of stockholders by shares, so participants' shares could not be watered without their consent. This may have adverse tax consequences and may become more complex to reduce the tax liability of this distribution.

All of this is a very complex way of implementing a simple idea—if one partner wants to work very hard for the company but has no cash at the moment, we can let him earn his share through "sweat equity", subject to the approval of the other holders. On the other hand, if a partner does not contribute the work he promised, his share will be gradually reduced as the other participants won't be likely to approve a stock dividend for him. Also, if a participant wants to increase his share by buying additional stock, he may do so at a price agreed to by the shareholders who may agree to sell it to him.

I want to make it clear that this is primarily a way to accommodate cases of hardship where the initial capital contribution is absolutely impossible to obtain at the start, and also to create an incentive for producing work as promised. It is not a way for all partners to avoid contributing capital to the venture—after all, those who do not contribute initially have no guarantee that they will ever be voted a stock dividend—they're trusting those who hold the majority share to compensate them when the time comes.

At this point it looks like if we can do it without adverse tax consequences we will go ahead and incorporate the venture. To avoid the tax disasters, we will remain a "software manufacturer" selling discs rather than licensing our products for a per-copy fee. As soon as we begin to generate revenue we want to pay out, we will put all the shareholders on the payroll, thus avoiding a large part of the double taxation of dividends. At this point the final word isn't in on whether we can make a limited partnership do what we want to do, so this decision has not been reached. We will be consulting a lawyer who has formed numerous high-tech ventures and who can presumably tell us what we ought to be doing. I'll send out another letter once we find out. I'm sending out this information at this time so you know what we're thinking at the present moment so you can comment on it.

New participants

We have already received participation commitments from two of our overseas contacts. Rudolf Künzli of Basel, Switzerland has extensive systems and applications programming experience and will be helping with software development and testing as well as marketing our products in continental Europe. His expertise in languages will enable us to offer products that stand out by not speaking English exclusively. Peter Goldmann in England has extensive experience in systems programming and data communications as well as the all-around experience common to those present at the meeting. We expect our dealer in London, Richard Handyside to become involved also in some capacity; we're pursuing several options at this time.

From our experience in MSL, we've found that the export market is very important, and I feel that these participants will give us an important start in marketing our products overseas, as well as market research and product customisation for these markets. Remember, the computer market in the EEC alone is the same size as the US domestic market. Ignoring it can cut your sales in half before you even start.[18] Also, since software is

[18] And indeed, Autodesk's sales outside the United States have accounted for between 30% and 40% of total sales for much of the company's history.

considered printed matter, it avoids almost all customs hassles, so it doesn't really matter where your customers are.

Resumes

As was discussed at the meeting, we'd like all the participants to send resumes to us. These will be kept as part of the MSP business plan, and are essential if we need to secure venture capital. Also, we'll copy all of them and send copies to everybody so we all know what skills we have in the company. What we want is more a statement of qualifications rather than all the job summary garbage. What matters is what you know, what you can do, and what you've done.

"Edges"

I'd like everybody to be thinking of things we can do to distinguish our products as a whole from other peoples', and give dealers and distributors reasons to try our products in the first place. Two have been suggested so far:

Rudolf Künzli suggested that we make all of our software obtain all its messages, menus, and prompts from a direct file.[19] We would develop a common routine which returns message text from the file by number, and a subroutine which inserts text in the message.[20] This gives us two important advantages. First of all, the most common customisation request for all packages is to change certain messages. We can tell the dealer, "Buy a MSP package, and you can change the messages with this little utility—no programmer is needed". Second, we can make our packages speak any language we want just by translating the message file—one object code version will suffice. The advantages in the overseas market are obvious. Note that a pure **PRINT USING** type expansion isn't quite enough—you'd like to be able to change the order things are inserted in the message. Thus, you might read a message like:

```
"Put the #1 in the #2, #3!"
```

and print it with something like (assuming QBASIC):

```
a%=fn.print.msg(187,"disc","slot","idiot")
```

The **#n** in the message would match with the order of parameters in the call (yes, I know the problems with this example—but you understood the *point*, right?).

Second, we can make our packages work on any terminal with no special generation required. Thanks to Greg Lutz, we've obtained a copy of the UC Berkeley terminal capability database, and Mike Riddle has written a program to decode it into easily accessed parameters. By writing a universal terminal module that is driven by

[19] Not until AutoCAD Release 12 was this done. It was the first major development project performed at the European Software Centre in Neuchâtel.

[20] This suggestion is essentially identical to the concept of "string resources" in the Macintosh. Apple subsequently invented string resources, the same way they invented the personal computer, the mouse, pop-up menus, and windows.

these parameters, a program can adapt to a terminal simply by taking the name of the terminal, looking it up the database, and plugging the parameters into the driver. As the UC database is constantly updated, most of the maintenance work is done by our tax dollars, not our flying fingers. If somebody shows up with a terminal not in the database, we still only have to make an entry in the file, not program a new driver. Thus, a dealer selling our products need only set up a configuration file when selling the package with a statement like:

```
TERM ZORCHTERM-100
```

and our package will be ready to go. I think this is a powerful selling point and we should do it for sure.

Now, what are your ideas? I don't want to jump into thinking of products just yet, but what are the company-wide concepts we should be putting into everything we do? Or, putting it another way, what things do you find most annoying on the system you use now, and how would you solve them if you were starting over?

Nightmare

On March 19, 1982, the West Coast Computer Faire will open at the San Francisco convention center. MSL has forked over $1200 for a booth at the show, and at the moment our only plans are to have an Interact system there. It would be very nice to show some MSP products at the show, complete with glossy brochures. Any ideas? At this point, I'm perfectly willing to cobble up things that look like products, which we'll clearly indicate are not ready for release. Remember that this is one of the major contacts between sellers and buyers and the only one in the Bay area. If you have any ideas, give me a call and we'll get cracking on it. There's only about 30 days left.

What's going on

At the moment, we're in the process of consulting with various people with experience in start-ups of this nature, and trying to line up marketing people. In a week or so we should know a lot more about what we can and can't do from the legal standpoint, and we'll try to put together a tentative charter which we'll send to you as soon as it's ready.

We're doing some market research, talking to people involved with Selector and other products to find out what their experience has been in this market. We're studying various desktop machines and thinking about how we can get the maximum development capability for our hardware dollar.

I think that progress is being made on all fronts, and at this point things look very good indeed.

What to Name the Company

Talk about an identity crisis! Virtually every name we came up with for the company was either considered harmful by the founders, or considered already taken by the California Secretary of State. Our numerous attempts to find a name didn't deter us from making ever more imaginative suggestions. First, a passel of names proposed by Duff Kurland.

To: WALKER
DUFF (02/26-11:15)

"Integrity Software" sounds good.... I did have a couple of other ideas, however (and they didn't even involve puns!)....

Desktop Software	Valu-Ware
Desktop Solutions	Future-Ware

I kind of like "Desktop Software", but can't help wondering where it will go: ..."Bottom Left Drawer Software"..."Shoebox Software"..."Breast Pocket Software"..."Shoestring Software"...(Hmmmm...). Other suggestions (some in jest):

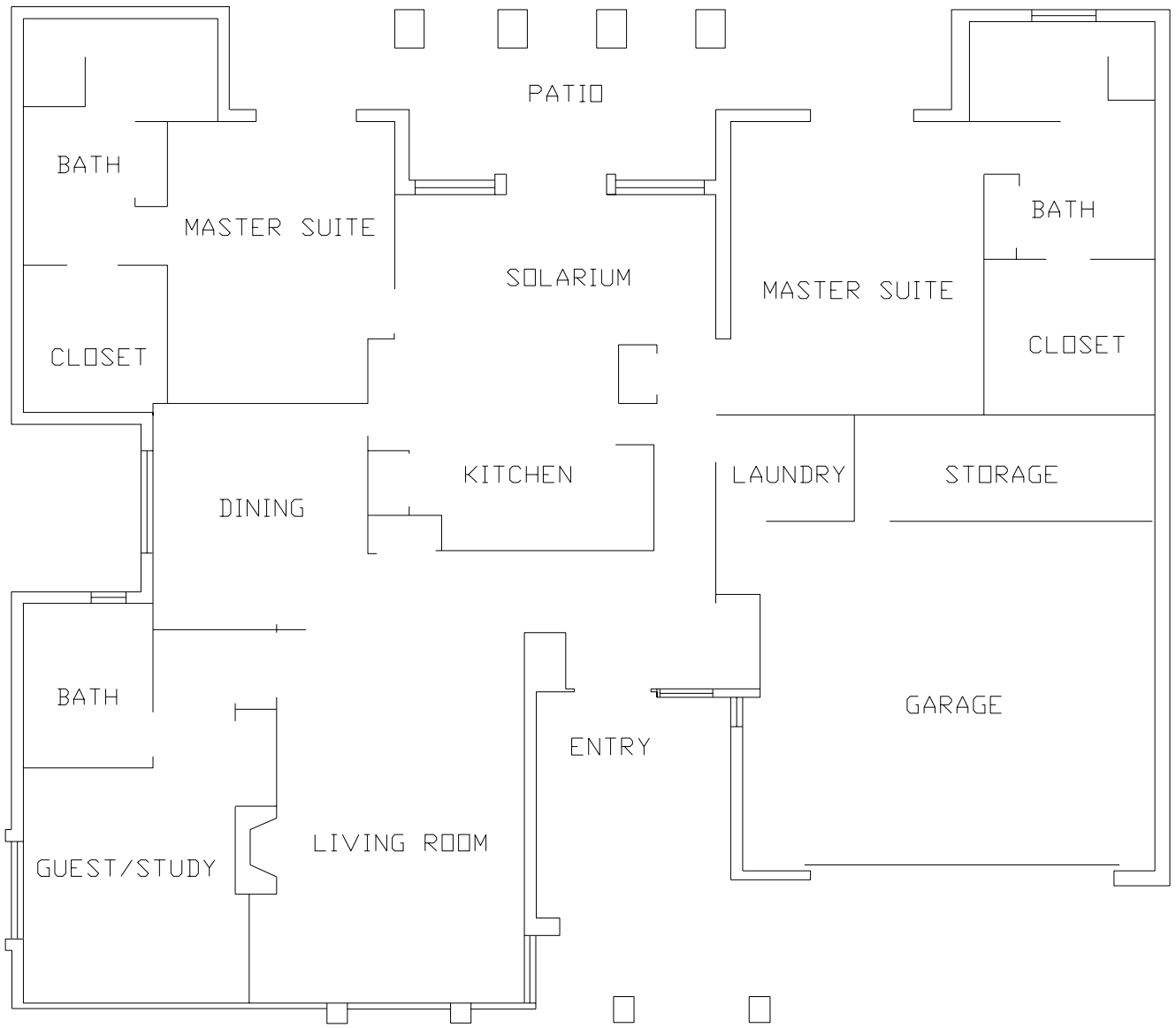
Office Solutions	Execu-Ware	Out-of-Control Data Corp.
Titanic Software	Business Ware	MIS Information Systems
Good OfficeKeeping	Manage-Ware	Ethical Ripoffs, Inc.
Dud & Brannstreet	Mr. Softee	Software Breakthroughs
Office Technology	Upper-Ware	Management Technology
Conceptual Elegance	Compu-Freaks	Smelly Rand

Other names proposed for the company with various degrees of seriousness included:

Command Technologies	Command Line Technology
RHT, Inc.	Target Software
Insight Automation, Ltd.	Coders Of the Lost Spark
Autodesk, Inc.	

Of these names, "Desktop Solutions" was initially chosen. We showed AutoCAD and the Autodesk prototype at the West Coast Computer Faire in 1982 under this name, and it appeared on the first brochures we ever printed. It was rejected by the Secretary of State, as were "Target Software" and "Insight Automation". By the way, "RHT" stood for "Red Hot Techies".

This drawing is another converted from M9900 INTERACT to AutoCAD-80 and thence to AutoCAD-86 by DXF. We showed it at COMDEX in 1982 and used it on the sample drawings disc for the first AutoCAD release.



(C) 1978 The Frank Lloyd Wright Foundation
used by permission

Information Letter 3

Information Letter 3 was the first to delve into the gory details of how the company was to be organised, capitalised, and run. Dan Drake wrote this information letter after consultations with Robert Tufts, a San Francisco lawyer to whom we were introduced by Jack Stuppin.

MSP Information Letter #3

by Dan Drake
March 2, 1982

The Organization Plan which is included with this mailing is the proposed plain English version of our plans. On Sunday we hope to reach agreement on the real thing, which we'll get written up by legal counsel. If we don't run into any snags at that point, we'll go ahead with forming the corporation.

At this point I ought stress that I am not now, nor have I ever been, professionally qualified to give financial or legal advice; I don't think that there are substantial errors of fact or law in this paper, but there may be.

Though this is a corporation rather than the limited partnership that we favored at first, the organization plan in general is very close to what we talked about on January 30 and what people expressed in their letters. There has been a change in the plans for getting computers, but the change should have little practical effect.

The business entity

The company is going to be organized as a privately held California corporation. In effect, the government wants us to be a corporation, and there is not enough reason to buck it.

The argument against a general partnership, in brief, is that any general partner can commit all the assets of the company. Furthermore, the general partners have to stand behind the company's commitments not only with their shares of the partnership, but with everything they own. The partnership agreement may name some managing partners who are the only ones authorized to act for the company, but the company could still be bound by unauthorized actions!

A limited partnership is hardly better. The law is not absolutely clear, but it is likely that limited partners who took any active part in the business would be declared general partners as soon as any litigation started, which reduces this to the previous case.

So a corporation it is. Here's a really crude outline of the procedure:

After registering the corporation with the state, the people who are doing the grungework appoint themselves as the Board of Directors and do assorted necessary paperwork. Part of this paperwork is a plan for the issuance of stock.

Then we hold a grand meeting at which we issue shares of stock in return for cash, notes, and other things. We immediately hold a stockholders' meeting to approve a stock option plan, and to elect a new Board if we want to. At that point we're officially in business.

The shares will be common voting stock, representing a fractional interest in the company, just like General Motors stock (though with a few little differences):

- Each share entitles you to one vote at a stockholders' meeting. This vote actually means something, which is more than you can say for GM stock.
- If the company folds up, the stock represents your cut of whatever is left over after paying the creditors. If that amount is negative, you're not liable for the difference beyond any amount that you may still owe on the stock purchase.
- If we sell the whole operation to another company or the public, the stock represents your cut of whatever is paid for the business assets. Employment by the successor company, of course, is a separate matter.
- You can't take your shares to your broker and sell them. It may even be illegal to sell them to your neighbor or to anyone else outside the company. Of course, after we're successful, we might go public and sell shares for a fantastic price like The Two Steves (Jobs & Wozniak of Apple). Even a public offering of new stock might not allow you to sell old stock publicly, but that decision would be up to the stockholders.

Issuance of shares

The basic arrangement for the first issue of stock is rather simpler than the things we talked about in January.

1. The stock will be issued at $1.00 a share.
2. If you have the money ready, you can buy any number of shares for cash. (There's an extra goody attached to this, described later.)
3. If you have computing equipment relevant to the company's needs, you can sell it for stock at fair market value. Obviously you don't want to do this if you're using the computer in a consulting business and don't want it moved out of town by the company. (If you've taken accelerated depreciation or investment credit, you'll have to worry about recapture on your next income tax form.)
4. You can get up to 3,000 shares on a 10% note, which we expect you to redeem out of your share of the income when there is any. If the company goes belly-up, however, you're fully liable for this loan.
5. Everyone is to take at least 3,000 shares on some basis or other.

You'll notice that we have written everyone down for some amount of stock in the Organization Plan. Don't be upset if you don't recognize the numbers opposite your name; we had to make some kind of guess, and this doesn't represent a commitment, expectation, or anything else.

We expect to issue some additional shares for other considerations. Among these will be the rights to Interact and the expenses that MSL incurs during the formation of the company. We may also sell small blocks of stock for cash to non-employees closely associated with the founding of the company, such as legal and financial experts.

Shortly before the stock is issued, we need to know exactly how much each person is taking, and on which basis.

Buying equipment

The plan is now for the company to buy whatever equipment it needs out of its own funds. If you already have equipment, you can sell it to the company, or you can go on using it instead of using a company machine. This gives the company control over the choice and allocation of equipment, and lets people invest as much as possible directly in stock. The disadvantage of this arrangement is that the tax breaks are less attractive, but tax breaks are only one consideration out of many.

Working capital

Out of the cash that we get for issuing stock we'll pay for equipment and the costs of setting up the corporation. This will leave us with enough money in the treasury to pay the very small expenses of the first few months, when we have no costs for salaries, rent, or advertising.

Once we have products to sell, we'll need much more money to carry us until we start getting enough income to cover current expenses. The number we've talked about is a total of $100,000 beyond the initial equipment purchases. To raise the money we expect to sell more stock during the first 12–18 months of operation.

The specific plan is to issue warrants along with the first issue of stock. A warrant does not convey any ownership share in the company, but entitles the holder to buy another share at a set price, namely $1.00. If it isn't exercised within a fixed time, it turns into wallpaper. It can be bought and sold on the same basis as stock.

The people who are expected to come up with additional financing (currently Marinchip Systems Ltd. and John Walker) will be issued warrants. In addition, everyone who is buying stock for cash will actually get a "unit" consisting of one share and one warrant, for $1.01; this is the extra goody, mentioned earlier, to encourage people to provide the company with cash. The warrants will probably expire in 18 months.

Stock options

One of the essential ideas of the company is a sweat equity plan by which people get an ownership interest in lieu of salary during the startup. Stock option plans are now a very attractive way of handling this.

Basically, the company issues options which can be exercised in the future at a fixed price. To qualify for tax breaks, this price will be 110% of the fair market value when the options are issued. What we hope is that the stock will become extremely valuable so that you can exercise your option at the cheap price, sell for a high price, and pay capital gains tax on the difference (plus straight income tax on the option price).

The tax laws force a few conditions on the price and expiration period of the options, but these should not be troublesome. Within these conditions we have a great deal of freedom in specifying the terms of the options. We'll circulate more detailed information on qualified stock option plans later, when we've consulted officially with the experts.

The Organization Plan includes an outline of a stock option plan. We ought to get a pretty firm agreement on details during the Sunday meeting, since this is such an important part of the whole plan.

Unpleasant question: What if someone does no work at all? In the extreme case he can be fired, forfeiting any options he has, but retaining any stock. In lesser cases he gets a severely truncated option.

Next unpleasant question: How is it determined who has been working enough? This has to be subjective; it can't be a matter of lines of code generated, divided by bug reports. The subjective judgement should follow easily from the experience of answering the phone and telling customers which products aren't ready yet. Inevitable differences in productivity are handled by bonuses for brilliant work and by not having duds among the founders.

Personal holding companies

One thing that scared us when we considered incorporating was the personal holding company rule, which can impose a 70% penalty tax on a corporation that makes too much of its money from royalties or other passive income. It turns out that there's a nice, clean exemption, designed for the use of movie and TV production companies, whose business is really very similar to the software business.

The rule is something like this: if half your income is from the sale of copyrighted material, and you spend 15% on expenses other than salaries, you're not a personal holding company. Our material will certainly be copyrighted, and we'll have no trouble spending 15% on advertising, so we seem to be home free.[21]

Conflicts of interest

There are potentially serious problems from a person's present employer claiming ownership of anything the person does for the new venture. If you have signed any agreement on ownership of patent rights, etc., please

[21] Dan Drake notes: In fact this exemption didn't apply; it was written explicitly for the film production business. Our special interest got its exemption only in 1986, when the IRS challenged Microsoft claiming that *all* of their income from the sale of software was royalty income and was clearly preparing to go after the rest of us. Thanks for picking on the most powerful victim first, guys!

get us a copy of it.

Even if you haven't signed an agreement, you have certain responsibilities to your present employer, if any. We'll have to have legal counsel draw up a paper by which everyone will make clear his right to create software for the venture.

New people

To strengthen the business end of this business we've enticed Jack Stuppin to join us. Jack has several years experience in running a company that manufactured mass-marketed products. He has even more experience in finance, including some startups of companies in the silicon business.

We also seem to have found an accountant and a lawyer for the company, both first-rate. The lawyer, Bob Tufts, has worked with Jack on high-tech business startups and has expressed interest in making a small investment in MSP.

Names, names, names

Speaking of MSP, the need for a name for this company has become critical! We also need names for the things that we're now calling Interact and Cardfile. It would be really nice if we could latch onto a neat little prefix, like Visi-, to distinguish our products. Please, please come to Sunday's meeting with a list of all the names you can think of, no matter how silly.

In the meantime, we'll probably have to print brochures for the Computer Faire, using Marinchip's logo and arbitrary names for the products.[22]

Paperwork

In addition to employees' agreements, we need some more information from everybody.

First, after studying the organization plan, please indicate as specifically as you can what your financial and working participation will be. The numbers in the plan are based on the first letters that people sent, but the figures in those letters were sometimes vague, and the rules have changed to some extent.

We also need your phone number and the name and address that you want entered in the company records.

And the resume. Don't bother with a fancy one, suitable for impressing employers, but give a good summary of your technical background. If we want to impress IBM with our qualifications, we'll re-write the resumes in a uniform style.

[22] We ended up showing the prototype of Cardfile, which had been renamed "Autodesk" and Interact, which was then called "Micro-CAD" in a booth with a company name of Desktop Solutions. Desktop Solutions was later rejected as a company name by California, so we had to choose again... and again... and again. "Autodesk"—snappy name that! MicroCAD was such a neat name for a CAD product that a certain fine gentleman who saw us using it at the show ran out and trademarked it out from under us, and used it for his own product. Success did not smile upon his product; the wheel of Karma turns slowly, but it is accurate to the least significant bit.

Next meeting

The next meeting is on Sunday, March 7, at 12 Noon. It will be at Jack Stuppin's house in San Francisco.

Does anyone have a good cassette recorder for recording this meeting? I'll bring a cheapo, but if you have anything decent, please call John at 383-1545 to volunteer as a recording engineer. We also need to make copies of the recording so that we can send them to several widely scattered people.

Proposed Organization Plan for MSP

This plan covers the initial issue of stock and warrants by MSP (the code name for the new venture). The purpose is to raise money for equipment purchases plus working capital for the first 12–18 months. The plan includes allowance for a qualified stock option plan.

Most of the numbers have been filled in, at least to a good approximation. Uncertainties are due to continuing negotiations and to the rough figures given in the participants' letters of intent.

Throughout the plan it is assumed that this will be a private offering, limited to the group named in the table of participants in a later section.

Initial Issue Of Securities

The company will issue approximately 100,000 shares of common stock out of an authorized total of at least 350,000. It will also issue approximately 100,000 warrants, expiring in 18 months, which can be exercised to buy common stock.

Stock will be issued in return for cash, computing equipment, notes, and other considerations at a price of $1.00 a share. Warrants will be issued at $.01 each.

Investors putting up cash will pay $1.01 for a unit consisting of one share of common stock and one warrant.

Those having computing equipment needed by the company will be able to sell it at a negotiated price representing fair market value, in return for common stock at $1.00 a share.

Each participant will be able to buy up to 3,000 shares on a 10% note payable in three years. The note is to be a recourse loan, representing a potential claim on any of the investor's assets.

Each participant is expected to invest in at least 3,000 shares by means of cash, equipment, or notes. The investment may be through a partnership or a corporation, though a corporation may not issue notes for shares.

It is expected that stock will be issued to Mike Riddle in return for the non-9900 rights to the Interact package, contingent on delivery of a working version for certain computers within a fixed time. Stock will also be issued to the founding group, Marinchip Systems Ltd. (MSL) in return for expenses incurred in organizing the new company and possibly for software, including QBASIC.

Stock Options

All or nearly all of the participants will be employed by the company at a nominal salary of $1.00 a year. In lieu of proper salary they are to receive stock options.

The first stockholders' meeting will adopt a qualified stock option plan covering all the participants who are employed by the company when it begins operations. There should be agreement in principle on this plan, including the quantities involved, before the corporation is organized. Here is a suggested outline:

150,000 (?) shares of the stock of this company (equal to 1.5 times the original issue of stock excluding warrants) are set aside for the company stock option plan, with options to be issued in equal quantities at the end of 6, 12, and 18 months after the company begins operation.

Each of the initial employees of the corporation will make a commitment to perform a specific amount of work per week for the corporation and will, having performed that work diligently, be entitled to options in proportion to that commitment. Smaller allocations, not necessarily in proportion to work performed, may be given to those who have not fully met the commitment.

In each distribution 60–75% of the options will be allocated according to work committed and performed, as described above. The remaining options will be awarded as bonuses for exceptional performance. The percentage allocated to bonuses need not be the same in all three distributions.

The Board of Directors will appoint a three-member Compensation Committee to determine the distribution of options. The Committee's plan will be submitted for approval of the Board, which may submit it to the stockholders. The resolution of the Board of Directors will set an option price which will be 100% to 110% of the current fair market value of the stock. Options will be valid for five years from the time of issuance, but will in any case expire upon termination of employment.

Liabilities of the Corporation

Out of the cash received for the first stock issue the company will buy about $25,000–30,000 of equipment and pay any remaining costs of organization. During the first 90 days of operation other expenses should be nominal, limited to telephone costs, printing of letterheads, and such.

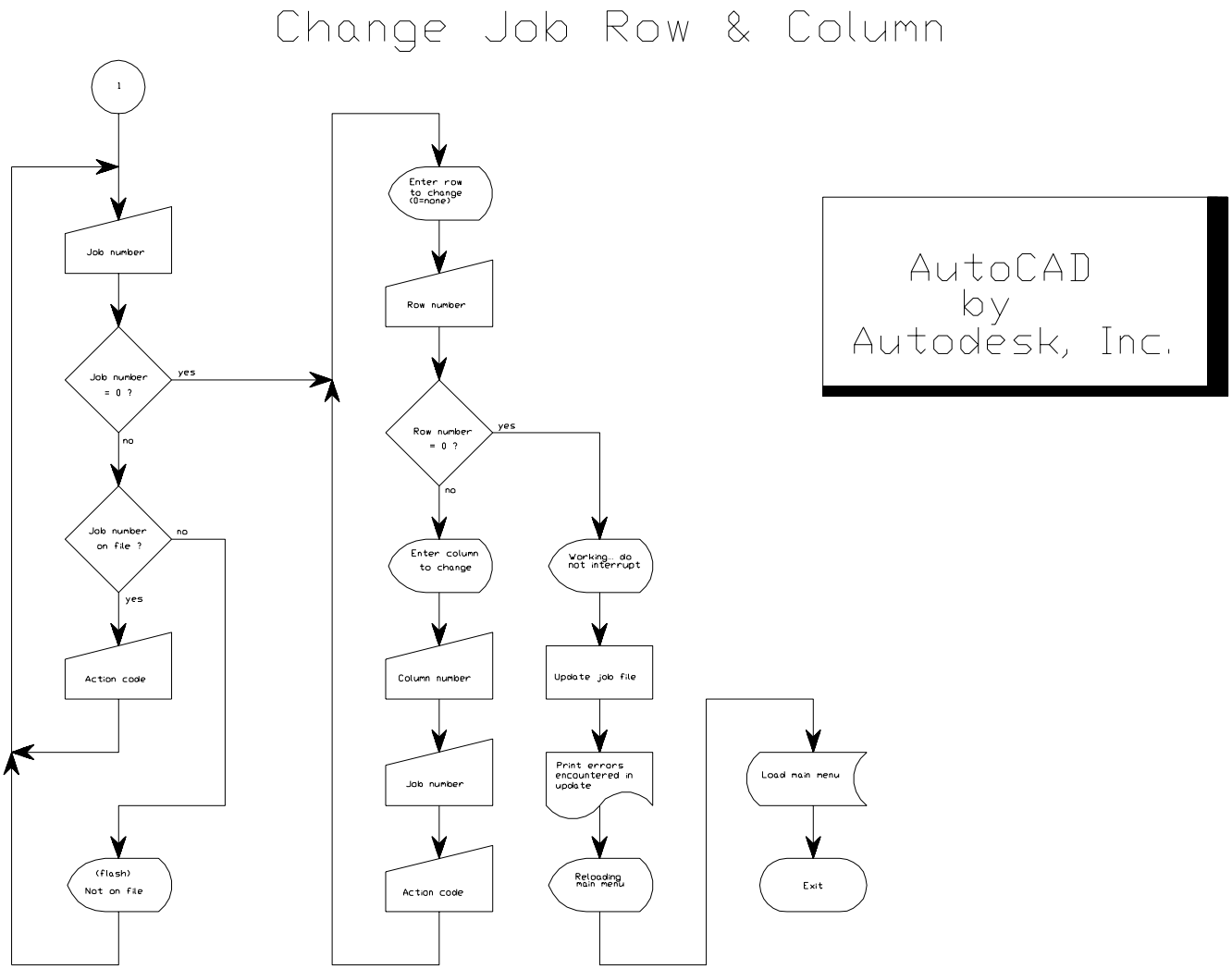
When products start to be available for sale, there will be expenses for sales and production. Until revenues match operating expenses, the company expects to raise operating capital by the participants' exercise of warrants.

In order to get the rights to Interact, the company expects to enter into a royalty agreement with Owens Associates, which underwrote some of the development of the package. The details of this agreement have not been worked out.

As most of the participants are now employed in the computer industry, there is a possibility of conflicting claims to the rights to software written for MSP. All participants will be required to certify that they have the right to develop software for the company, clear of any claims by any other employer.

No other liabilities are known.

This drawing was originally done on AutoCAD-80 shortly before COMDEX 1982 as a show demo. The program described by this flowchart is one of the Marinchip business application packages.



Information Letter 4

Information Letter 4 followed Letter 3 by a month, and brought the participants up to date on the organisation plan as it stood after being processed by the lawyers and the California Commissioner of Corporations. Shortly before this letter was written, we had showed prototypes of the Autodesk product and MicroCAD, the product which ended up being called AutoCAD. Dan Drake is again the author.

MSP Information Letter #4

by Dan Drake
April 2, 1982

This letter summarizes what has happened since the last letter and what we expect to do next. The plans are based on what we think is the consensus of all the people who have expressed an opinion. Now that we've recovered from the Computer Faire, we're going to move ahead as fast as possible, so speak up if you find anything wrong here.

New organization plan

Dan Drake, Keith Marcelius, and Jack Stuppin met on March 15 with Robert Tufts, who will be the attorney for the corporation, to review our plans for setting up the corporation. Nothing was fundamentally wrong, but there are some serious regulatory problems that have forced some changes in the plans. Here is the new organization plan:

The initial offering of stock will be to the 13 founders who are legal residents of California. These people will buy the units of one share and one warrant, as described in the last letter, in return for cash, equipment, or accounts receivable (in the case of Marinchip Systems Ltd). The company will then hire employees (us) and offer up to 3,000 shares of stock apiece in return for 10% notes (by law, only employees can buy stock for notes).

After all this is done, the company will hire more employees, namely the people who aren't residents of California. As an incentive to join the company these people will get stock options which will put them on essentially the same basis as the original employees.

There won't be any special issue of warrants beyond the one-for-one deal with shares of stock, because that

could raise tax problems. However, we expect to offer a large option to the president of the company, John Walker, to give him an incentive to commit his time and capital.

General meeting

At the March 16 meeting we went over all these legalities and discussed some questions that Bob Tufts had raised. The rest of this section gives the decisions that we reached.

Though we are capitalizing the company in a way that saves us from meeting the fantastically expensive requirements of the Securities and Exchange Commission, there remain the less stringent requirements of the state of California. We have the choice of applying in advance for a permit to offer stock or simply filing notice of a private offering with the Secretary of State. The former slightly reduces the risk of later legal problems at a cost of $700–1,000 and 2–3 weeks delay. The consensus of the meeting was that it's not worth it; we'll sell the stock, then file the notice of private offering.

Private corporations often have special agreements that prevent stock from getting into the hands of outsiders. In order to cover lots of contingencies, including death, divorce, bankruptcy, and other involuntary transfers of stock, these agreements get very long and messy. Our decision was that a fancy agreement is not worth the time and expense; we'll just make an agreement that the company and its stockholders have first refusal if anyone wants to sell his stock to an outsider.

MSP obviously doesn't want to force its employees (ourselves) to sign the usual employee software rights agreement, with the usual restrictive and unenforceable clauses. We'll write our own agreement that says: I have the right to produce software for MSP without a prior claim by someone else; I won't use other people's trade secrets; I won't steal trade secrets from MSP; I will give MSP first refusal on any ideas I develop for mass market software. Under the last clause you can write anything that's not for a large market; if it later develops mass market potential, you give MSP first refusal on it.

We have collected the full legal names and addresses of nearly everyone in the company, as needed for company records, stock registration, and whatnot. The list is given at the end of this letter. If your name is not on it, please give us your full legal name, with parentheses around parts of the name that are not normally used (!), and your address.

Finally, the Board of Directors of the company will have 3–5 members, an item that must appear in the articles of incorporation. Though nothing has been officially determined, it is likely that the board will consist initially of the people in the north bay area who are actively in touch and have time to devote to organizational trivia: Dan Drake, Keith Marcelius, Jack Stuppin, and John Walker.

Plans: Incorporation

At the March 16 meeting we reached a pseudo-consensus on an unsatisfactory name for the corporation (Autodesk Inc.), but by the weekend the consensus seemed to have fallen apart. On this crucial question no one is satisfied, but everyone feels burned out. So we're going to file the Articles of Incorporation in a few days, using the name INSIGHT AUTOMATION LTD unless (1) the state disallows it[23] or (2) someone comes up

[23] They did, just as they had previously disallowed "Desktop Solutions", the name under which we showed at the Computer Faire.

with the perfect name in the next 5 or 6 days.[24]

Technical progress

John Walker made an impressive demo version of Autodesk, our super filing system, for the Computer Faire. It got quite a good response, especially considering that we didn't claim to be able to release it in less than three months.

John also has the Z80 slave processor from Sierra Data Systems running CPM 2.2 under supervision of our 9900 system. This means we can install CPM in any of our systems for about $600. He has also converted WINDOW to CB-80, the compiled version of CBASIC. All we have to do is convert 1,000–2,000 lines of assembly code, which will allow it to page files on disc, and we have a valuable product for CPM systems ready to go![25]

Technical plans

If only because of the support burden, we can't target every computer system in the world during the first few months. The current idea is to pursue the CPM (8080 and Z80) market immediately with all we've got. This means installing the Sierra Z80 board in lots of existing computers.

We need to do more evaluation of the IBM and Apple situation with respect to both technical and marketing questions. We ought to be getting hardware for non-Z80 systems within 4-6 weeks.

The products that we expect to concentrate on are MicroCAD, Autodesk, Opticalc, and Window. One non-yucky name in the whole bunch. For those who haven't kept up with the latest nomenclature and bright ideas, here's a quick description.

MicroCAD: The new name for Mike Riddle's computer graphics package. We've printed brochures for this, so the name is pretty well committed. Naturally, this won't be on 8-bit machines; we're hitting IBM first.

Autodesk: The instant filing system that will knock Visidex etc. out of the running. This too has a brochure.

Opticalc: The name we've been using for a brilliant idea that came up during the Faire: a VisiClone (spreadsheet package) that performs some optimization on its own.

Window: Marinchip's screen editor, converted to CPM systems.

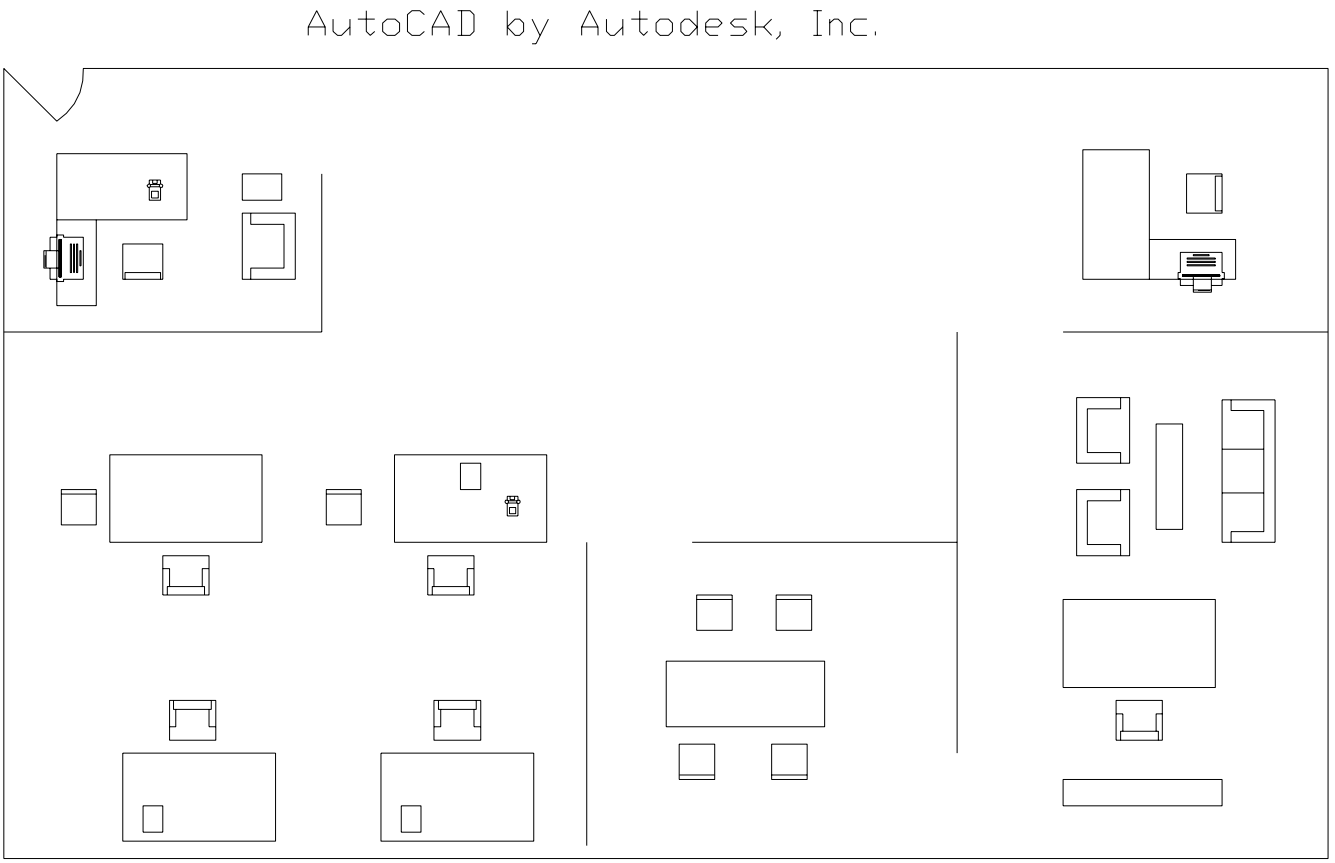
[24] Nobody did. Suggestions included "Coders of the Lost Spark". See page 47 for more.

[25] This product eventually came to market as AutoScreen. It was introduced at COMDEX 1982 at the same time as AutoCAD. We sold about 10. The current Autodesk programmers' editor "Kern's Editor" is a descendent of AutoScreen.

We're also looking closely at JPLDIS, a very useful data base system written in Univac Fortran. The program is in the public domain, so we have the right to convert it to microcomputers and sell it. In fact, it apparently is being sold now under the name of Dbase II, but there's nothing to stop us from getting into the act.

We still need to make decisions about an implementation language, for which the candidates seem to be CB-80, C, and PL/1. We now know that CB-80 works, and we can expect it on the 8086 sometime soon. PL/1 also works, and we can get a beta test version in May. C is supposed to be available on every microcomputer; we need to know more about the quality and standardization of the various versions.

The office drawing was done in the week before COMDEX 1982 as a flagship demo to show off zooming and block manipulation capability. It was the most-used demo at the introduction of AutoCAD.



Initial Stock Distribution

Autodesk, Inc. was officially incorporated in California on April 26, 1982. This letter accompanied the distribution of the company's stock to the founders. The original shares in the company were sold for $1 each. Five years later, adjusting for stock splits, each of those original shares had appreciated in value to more than $1100. This is why people start companies.[26]

Autodesk, Inc.

Greetings,

Enclosed are your stock certificate(s) and other Autodesk related documents. This letter should answer some of the obvious questions about them.

"Why did I get two stock certificates?"

If you purchased stock for cash, and also for a loan (note), separate certificates were issued for the two purchases. This purely to simplify bookkeeping; the effect is the same as if you had one certificate with the sum of the number of shares on the two. If you purchased shares only for cash or only for a note, you will get one certificate.

"What's the 'Combined Intrastate and "Private"' Document?"

This is your copy of the Investment Letter which you signed before purchasing your shares. As the letter is keyed to the particular number of shares in a given certificate, if you got two certificates, you'll also have two investment letters (check number of shares in letters to see which is which). The originals are on file here; this is your copy for your records.

"What about the note for shares?"

You should have received a copy of the Promissory Note for the shares purchased for a loan when you signed the original. The copy is stamped "COPY" in red. If you cannot find this copy, we can make another and send it to you.

"Could you explain the warrant again?"

[26] By mid-1991, each original share of Autodesk stock was worth $2700. This is why investors buy stock from people who start companies.

If you purchased stock for cash, you also purchased warrants to acquire an equal number of shares to those purchased for cash. The price of the warrant was 1 cent per share. In the package is the original warrant you purchased. This warrant is valid for four years. At any time during the 4 year period, you can exercise your right to purchase any number of shares from 1 to the total number listed on Page 1 of the warrant, by paying $1 for each share (regardless of the price or value of the shares at the time you purchase the warrant). The warrant is exercise by returning Page 8, "Subscription Form", with your payment to the company. As explained in that form, if you purchase less than the total number of shares purchasable under the warrant, you will be issued a new warrant for the balance of the shares still unpurchased at the time the shares are delivered. Or in other words, you can buy any number of shares any time you like in any size chunks you want, up to the total listed in the original warrant. The warrant is a piece of property just like a stock certificate, and Page 7 is used if you sell it to somebody else (note that sale may be subject to restrictions under law and the bylaws of the company, just like a sale of stock).

"What do I do with this stuff?"

The stock certificates and warrants are your physical property, and are the tangible evidence of your ownership in the company and right to increase it by exercising the warrant. Don't lose them! They are *very* painful to replace. If you have a safe deposit box, that's where they should be. The rest of the documents are for your files relating to the company, as they are your copies of documents for which the company holds the originals.

"What do I do with the receipt?"

Included are two copies of a receipt for these documents. After you verify that the documents described in the receipt have been delivered, please sign and return the copy marked "sign and return". I've enclosed a SASE for people I'm not handing the shares to personally. An additional copy of the receipt is included for your records.

Call me if I've forgotten to mention something.

Sincerely,

John Walker

Information Letter 5

Autodesk, Inc. was officially incorporated in California on April 26, 1982. I wrote this information letter just after finishing all of the paperwork associated with the incorporation, including mailing stock certificates, options, and warrants to all of the founders.

Autodesk, Inc.
Information Letter # 5

by John Walker
Revision 9 — May Day, 1982

This letter is to bring you up to date on the progress since the last letter, pass on some additional information about products with suggestions of who does what and some random company and product notes. If you have not already received one or been sent a copy, this letter will be accompanied with a copy of the "Autodesk Design Guide", the working paper for the final version of the Autodesk database system. Comments and suggestions regarding all of the enclosed are solicited.

Progress Organising the Company

Since every other name for which there was a general consensus of acceptability was unacceptable to California as a corporation name (due to being too close to somebody else's), the company ended up being called "Autodesk, Inc." (AI). If we think of the perfect name, we can always change it. In the meanwhile, I'll use "AI" when referring to the company and "Autodesk" when referring to the product with the same name.

The corporation was created by filing the Articles of Incorporation with the Secretary of State on April 9, 1982. The officers and board of directors were elected on April 16, 1982. The first phase of organisation of the company was completed on April 26, 1982, when the stock was sold to the founders (all California residents, as described in "Information Letter #4"), and the notes for stock purchased by loan were signed. The money from the stock sale was then deposited in a new account for AI opened at First Interstate Bank. Thus, the company is now officially formed and operating. Officers of the company were chosen as follows:

President: John Walker
Vice President and Secretary: Dan Drake
Treasurer (CFO) and Assistant Secretary: Keith Marcelius

Assistant Secretary: Bob Tufts

The doubling up of offices permits critical documents which require signature by "(President or Vice President) and (Secretary or Assistant Secretary)" to be filed when one of the officers is unavailable. Designating our attorney, Bob Tufts, as Assistant Secretary allows routine matters which require signature by a Secretary to be handled without getting all the officers together. This form of organisation is typical of companies of our type.

The Board of Directors is as follows:

Dan Drake
Keith Marcelius
Jack Stuppin
John Walker

As mentioned in Information Letter #4, the board consists of those who are willing to get together and deal with the matters the board is required to deal with. There were no other volunteers, so the board is as suggested in that letter.

Neither the slate of officers nor the board membership is cast in concrete, of course.

Changes from Original Organisation Plan

Legal requirements forced several changes in the original Organisation Plan (OP) of March 2, 1982, which was sent with Information Letter #3. As far as I know, the following list of changes is all-inclusive. None of these changes significantly affects the status of any participant in the company, and none has any adverse affect we've been able to think of.

In the OP the warrants were stated to expire in 18 months. This has been changed to 4 years to give the recipients of the warrants more flexibility in deciding when to convert them into shares. The stock options issued to the out of state people will have the same term.

In the OP, the specifics of the stock option issuance were spelled out in an agreement in principle. The stock option plan finally adopted will be as stated in the OP. At this point, we have in effect a "legal boilerplate" option plan which allows the flexibility needed to accommodate the plan described in the OP, plus the ability to bring in the out of state people via options as described in IL #4.

Jack Stuppin is an employee of a member firm of the New York Stock Exchange, and is hence prohibited from being an employee of any other company. As a result, he cannot be an employee of AI, but he can be a director. He can still work for the company in the capacity of director. Since he's not an employee, we can't grant him stock options like the other employees, so we issued him warrants as a director.

The only other change from Information Letter #4 is that Marinchip Systems Ltd. was issued warrants for the stock it is to purchase over the next year. In IL #4, MSL was listed as purchasing stock for a note. Since only employees can purchase stock for a note, and a corporation can't be an employee, we accomplished the same effect by issuing warrants. (Actually this way is slightly better: this way, MSL does not gain a vote in the operation of the company until it comes up with the money. Had we issued it stock for a note, it would gain

the votes immediately.)

What Happens Next

In terms of organising the company, the next step is to hire the out-of-state participants and grant them the stock options which they will use to effect their purchases of stock. Two options will be granted to all out-of-state participants: a 60 day option for initial purchase of stock, and a 4 year option equivalent to the 4 year warrant sold to California residents. Out-of-state participants will be given a note to sign to exercise the 60 day option identical to that signed by the California people. The 60 days in the option are simply to give people time to get the money in our hands, as this can take a while for international transfers.

Once the 60 day options have been exercised, everybody (California and out-of-state) will be on an equal basis, the company will be running, and then we'll hopefully be able to stop worrying about the form and get to work on the substance—developing products and selling them.

Just a Few More Little Things

Included with this mailing is a W-4 form for tax withholding which, even though we are paying salaries at the staggering rate of $1/year, we must have from all employees. We don't yet know how this works for overseas people, so if you're one of them just keep the form for the moment until we find out from the U.S. tax people. U.S. people, please fill out the form and return it to us in the enclosed SASE.

If you are a non-California resident, you will also find enclosed an employee agreement. This agreement has already been executed by all the California people. We must have this agreement as evidence of employment before we can issue the employee stock option that you will use to acquire your original share in the company. A duplicate agreement is enclosed for your records.

Bob Tufts has asked us to inform all participants that if they feel that their interests are being violated by the corporation, they are urged to contact their own legal counsel. This information was passed on to all people in the initial offering, and I am repeating it here for completeness.

Regular Meetings

To avoid the last-minute panic meetings which have characterised our operation through the organisation period, we've set up regular monthly meetings on the first weekend of every month, with the first meeting in June 1982. On even numbered months, the meeting will be on Sunday, on odd numbered months, Saturday (since neither day was preferable to everybody). Hence, the next meeting is Sunday, June 6, 1982. The location of each meeting will be set at the preceding meeting, and will be at various places to share the travel burden among the participants. The next meeting will be at Jack Stuppin's house in San Francisco (directions are in Information Letter #3).

The agenda of these meetings will be a review of company progress, product status, and other matters as described in the original Working Paper. We'll try to work the meetings so that technical sessions about

specific products can be worked in at the end to aid people who are collaborating on products in getting together.

Purchasing

In the process of evaluating products and getting the work done, participants will need to buy manuals for various products, supplies for machines provided by AI, etc. Our intent is that manuals you need to evaluate products, etc., can be handled by a petty cash mechanism—you buy it and send the receipt to AI, which will refund the money to you. If in doubt, call and ask. (In some cases, we know places to get things cheaper than the local computer store or direct from the manufacturer. Also, somebody else might have one you can borrow.) To avoid disputes, it's wise to clear everything first, but something like a $35 manual for a package competitive with one we're developing will naturally be O.K.

By combining AI's orders with MSL's ongoing business, we can get floppy discs and other computer supplies at very low prices. We will centrally purchase these items and ship them out to people as requested. We'll try to keep reasonable quantities of all the normal supplies in stock here, so Bay Area people can get what they need next day by UPS. Shipping costs make this crazy for overseas people, so we will reimburse them for supplies expenses from receipts they submit.

Micro-CAD Progress

We now have a formal agreement with Mike Riddle for transfer of MicroCAD to the company in return for royalties to Mike to compensate him for his personal development of the package. Thus, we're now on the way with one of our major products. Mike has been working with a CP/M 86 system he owns (on a Godbout 8085/8088 board) and has been using an IBM Personal Computer at the local Computerland store on off-hours. He has acquired the Microsoft Macro Assembler and tested it, and has determined how to interconvert programs between PCDOS/MSDOS and CP/M 86. This means we can develop on one and sell on both.

Current thinking is that our best path to getting MicroCAD running on the 8086/8088 is to port SPL and recompile with it. Now that we have an assembler, we have all the key tools in hand to move the META and SPL complex. Although we will have to do a substantial amount of 8086 assembly code to move the package, when we're done we'll control the compiler, and that allows us to take advantage of floating point chip options, extended memory beyond what IBM's operating system supports (which you can buy off the shelf today), and other such selling points much faster than if we had to wait for our language vendor to get around to supporting them. One of the major hassles in using the 8086 is the memory segmentation architecture to address large memory spaces.[27] Controlling the compiler allows us to be sure we won't be limited by the language to less memory than the machine allows.

To this end, Mike is plugging away on the SPL port. Once we get our IBM machine, we'll be able to increase the pace of this effort.

[27] Indeed. . . .

Autodesk Progress

At the moment, the Autodesk package is being designed by Kern Sibbald, Keith Marcelius, and John Walker. The attached copy of the design guide, which is the starting point for this design, is for your review. Please forward any comments and suggestions to one of the people mentioned above. If you want to get involved in the product design, you're welcome to. There's plenty left to do.

We're planning to implement Autodesk in CB80 for CP/M. We've worked enough with CB80 so far to trust it to hold up for a project of Autodesk's size. Also, since the prototype system developed for the Computer Faire was written in QBASIC, some parts, such as the large screen editor module, can be lifted and used essentially as-is.

Other Product Progress

The status of all of the other products is as listed in the enclosed products note. I think it's making more and more sense to look at QBASIC as a product because of the expected long delay before Microsoft delivers their true compiler for the IBM PC, the expectation that CB86 (whenever it is ready) will run only under CP/M 86, not PCDOS, and for our own use in porting our CBASIC products to the PC and other 8086/8088 systems. Also, as noted in the products bulletin, if we port SPL, we'll have done a lot of the work needed to port QBASIC already. We'd also be in an excellent position to move it to the 68000 and ace Digital Research out of the CBASIC market.

Product Polish and Packaging

In the mass market, the initial impression made by the packaging of the product (including the format and quality of its documentation), user aids, and dealer training tools. As you look at other products, try to keep these things in mind and note any good ideas you see or hear about from others. Products like Autodesk and MicroCAD need to look as professional as their operation is, and we need to think about what we should be doing in that way as we develop the products for initial marketing.

Digital Research can get away with crummy manuals because everybody knows them and respects their products. We can't because nobody will have heard of us, and if our manuals and packaging look amateurish then potential customers will assume our products are as well.

Two questions I'd like to ask everybody are:

> "What is the best software manual you've ever used?"
> "What made it so good and so useful?"

I'm sure we can produce manuals as good as any in the industry. But first we need to decide what we want to make. Let me know your thoughts.

Subchapter S Election Alternative

We have been assuming since IL #2 that we lacked the option to organise the company as a Subchapter S corporation because any corporation with nonresident alien (i.e. overseas) stockholders is ineligible for Subchapter S. It turns out that because of the way we're bringing in the non-California people, there is a reasonably attractive alternative we should consider (I will studiously avoid using the word "option" here except when I mean "qualified employee stock option"—hence all the "alternatives" herein).

First of all, what is a Subchapter S corporation? A normal corporation has assets and liabilities just like an individual. When the corporation makes money, the only way it can get it out to the owners is by paying dividends, which are taxed twice, or by paying salaries. More important in the case of start-up companies, if the corporation loses money, the losses simply reduce the net worth of the corporation; they cannot be used to reduce the stockholders' tax liability (but they can be carried forward and used to offset the corporation's future profits). (However, if the corporation goes totally belly-up, the stockholders can deduct the loss on the then-worthless stock.)

A Subchapter S corporation works very much like a partnership for tax purposes while retaining the limited liability and flexibility of a corporation. The net profit or loss from the corporation is simply divided among the shareholders based on percent ownership and declared on their tax returns on Schedule E as regular income. Since most corporations lose money in the start-up period (while you're doing the development, writing off the equipment you bought, and doing initial advertising), a Subchapter S corporation can pass these losses directly out to the people who, after all, put up the money that's being lost, so they can reduce their income taxes. If the company loses, say, $20000 in the first year and you own 5%, that means you can deduct $1000 from your income. If you're in the 35% marginal tax bracket, that means you keep about $350 rather than giving it to Uncle Sam. Good pay for filling out a form.

If the company starts to make money and you decide you don't want to be Subchapter S any more, you can change to a regular corporation. Once you've done that, you can't change back to Subchapter S for 5 years.

One catch in Subchapter S is that California law doesn't recognise it. That means that earnings are double taxed in California. But remember that the California top tax bracket is only (*did I say only?*) 11%, and besides it's better to save Federal taxes anyway, even if you don't save on California.

O.K., now that we all understand what Subchapter S is, how can we go with it even though we have overseas participants? The thing that makes it possible is that the overseas people are acquiring their stock through employee stock options, not through direct purchases. The law says that if you have a foreign stockholder, you're ineligible for Subchapter S, but there's nothing wrong with granting an option to somebody, as long as it isn't exercised. As soon the holder of that option sends his money and says, "Send me the stock", you're immediately bounced out of Subchapter S, but up to that time it's fine.

Now the plan so far has been that the non-California people would get 60 day options, which they would immediately exercise for their initial stock purchase. The alternative is to make the initial options for, say, 2 years, not exercisable until either 1 year has passed or the company has dropped Subchapter S. Mike Riddle, who's out of state but not overseas could still receive a 60 day option and exercise it immediately—his case is irrelevant to this discussion because it's only overseas shareholders which cause the Subchapter S problem.

Now what would this mean for the domestic shareholders and the overseas people? The domestic shareholders would be able to deduct the initial losses by the company from their taxes. As most of the participants have other jobs and are in reasonably high tax brackets, this would result in substantial reductions in their tax bills.

The overseas participants would be able to defer their initial stock purchase in the company, keeping their money until the option exercise time began. This would mean that they would not have to come up with the money right away, and if the company collapsed, would not be out the amount of the initial stock purchase, as they could let the option expire unexercised. If the company becomes successful, the option guarantees them their share at the initial offering price, so they can buy in on the same basis as the domestic people.

Let's look at the disadvantages. The domestic shareholders would be the only people putting up money immediately, so the company would not have access to the working capital generated by sales of shares to overseas people. The overseas people, once they exercised their options, would not have the prior losses of the company to reduce the corporate taxes paid on the (we hope) current profits of the company.

I think that in terms of financial benefits and disadvantages that this alternative is reasonable. The domestic people get to take advantage of tax benefits which wouldn't otherwise be available. The foreign people lose some benefits, but are compensated by having a year's use of their money before having to make the initial stock purchase, plus eventually owning a piece of a company whose initial capitalisation was done by the domestic people.

On the other hand, I think that this may simply be so confusing and hard to analyse that maybe we'd just be better off paying the money to the feds and getting to work writing software rather than further complicating the structure of the company. I personally feel very ambivalent about this matter. I certainly don't want to do anything which would make either the domestic or the foreign people feel like either was taking advantage of the other, so if anybody at all is concerned about it from that aspect, I think we should just forget it.

If we want to do it, here are the basic constraints: we have 75 days from April 9 to elect Subchapter S, and to do so we have to file a form with signatures of all stockholders and their spouses. Maybe we should start circulating the form just in case, so we have it if we decide to go ahead. Next, we need to get letters from all the overseas people giving their assurance that they understand the arrangement and that they approve it. This is required because they are foregoing the loss carry forward which reduces their eventual payouts on the stock (even though they are presumably compensated by not having to put up the money right away).

It seems to me that this is something that requires absolutely unanimous consent by everybody involved, so if you don't like it, let me know (and why). If you don't understand it, I'd be glad to "clarify" it at greater length (although at the present rate, I don't know how many pages that might take).

Autodesk Products

A document describing the status of products currently under implementation and potential products was mailed with Information Letter 5. Here it is.

Autodesk Products

by John Walker
Revision 4 — April 28, 1982

This note describes planned and potential products of Autodesk. The products are listed here in no particular order.

MicroCAD

The product is a computer-aided design and drafting system. This product currently exists on the Marinchip 9900 computer in SPL. Our plan is to convert it to the IBM Personal Computer either by translating it to C or Digital Research PL/1, or by porting SPL to the 8086. We would also be able to offer a very high performance system using the Godbout 10 Mhz 8086/8087 S-100 system.

Installed on a desktop computer configuration in the $10K to $15K range, it is competitive in performance and features to Computervision CAD systems in the $70K range. There are no known competitive products on microcomputers today (although there are some very simpleminded screen drawing programs for the Apple, and we must be careful to explain how we differ).

We can probably obtain substantial free publicity by issuing press releases and writing articles stressing the tie-in with computer aided design and the IBM robot controlled by the IBM personal computer. We can also aim our ads to sell the product as a "word processor for drawings". Potential customers are anybody who currently produces drawings. Small architectural offices are ideal prospects.

We can make add-ons to the package to make it an engineering workstation. It could be used to enter, edit, and view structural engineering information, for example, or to interactively view and edit plots made on mainframe computers and transmitted downline.

The package has been sold as a software package on the M9900 at $1000. There are about 20 installations at present, no substantial promotion has been done.

Autodesk

Autodesk is an office automation system for small computers. It embodies a computer model of an office environment. It provides file cards, file boxes, a calendar, etc. This is connected to a very simple database and query system. The entire system is intended to be extremely user-friendly and straightforward, We want a computer store salesman to be able to get an off-the-street prospect who's never seen a computer before to be using the package in 5 minutes or less. That will help him sell hardware, and he'll sell our software with it.

This package has an almost unlimited potential for add-on products for installed customers. Electronic mail, fancier filing, report writer modules, data entry systems, etc., can all be sold as add-ons to the basic system.

A prototype of the package written on QBASIC on the M9900 exists. Some of the largest sections of this can be used as part of the final product if CB80 is used on CP/M.

Based on the pricing of "card file" systems with much less capability, I would assume the system would sell for from $250 to $450 retail. This is a product which would probably sell very well on the Apple, if we could get it running on one.

Window

This is a screen editor written in QBASIC currently running on the M9900. About 100 to 200 are installed currently. A version converted to CB80 exists on CP/M, which needs only a line database module to be converted to be complete.

This product can be sold on CP/M as a competitor to VEDIT, probably the dominant product of its type. I think we'd have to bring it out for, say $99 to establish it. We want to price it in the "impulse purchase" range by the standards of computer stores. I don't think any substantial development work or extensions are in order here. This is a quickie product to get some money coming in.

Diff

This is a file compare program written in QBASIC. With about 3 days work it can be cleaned up and converted to CP/M. This is a very useful software tool not generally available on CP/M systems. I think we can sell it for about $50, probably directly. Its value is getting some publicity and generating some fast revenue. It could be expanded into something like the Unix source code control system if it surprises us and becomes a best-seller.

We can also offer a version that knows how to compare WordStar files. It could insert commands for change bars, and provide this very handy feature to WordStar users.

Opti-Calc

This product combines one of the most useful management and planning tools, linear programming, with the most popular user interface for planning, Visi-Calc. Visi-Calc allows you to specify dependencies of items and ask "what if". Opti-Calc lets you point at numbers and say to hold them constant, point at other numbers and say to maximise or minimise them, then it will set up the LP problem, solve it, and display the numbers changed as a result of the optimisation. This would make our product stand head and shoulders above all the competition, and properly handled should sell as fast as we can make them.

Other than the concept above, no work has been done on this product. Somebody who knows LP needs to design this thing after first gaining an in-depth understanding of Visi-Calc.

Communicator

One of the best selling utilities today is telecommunication software. Kern Sibbald has rewritten our TS utility from the 9900 in QBASIC, so we can convert it to CB80 for CP/M. To make it in the CP/M world we need to add lots of silly little features such as being able to do a directory within the program, etc. We'll also need to make it store a database of people, auto-dial by name, etc. My idea for how to distinguish it is to make it programmable in a simple language so you can preprogram dialogues for access to remote systems. That way you could code up a query access dialogue like the Apple Dow Jones program for any system and easily change it when the system changed. We would of course include preprogrammed dialogues for the most popular database services. The dialogue setup should be very simple-possibly something like PILOT might work—it's essential that an end user can develop dialogue scripts without calling us.

JPLDIS

JPLDIS is a public domain information management system written in Fortran by JPL. A re-implementation of it in assembly language called DBASE II[28] is selling like hotcakes at $800 a pop. We can get the Fortran source from COSMIC, convert it to something, and have a product. Somebody needs to get the sucker and see what we're in for here. It's a lot of FORTRAN, but if it will compile right through Microsoft FORTRAN, we may have a fast product.

QBASIC

Mike Riddle reports that Microsoft's true compiled BASIC for the IBM PC is at least a year off. We can take advantage of large chunks of the work done to port SPL to the 8086 (if we go that way for MicroCAD) and also port QBASIC. This would let us slip in with an efficient version of the most popular business BASIC on the hottest new machine. It would annoy Digital Research immensely, but we're not doing anything wrong as our implementation has no connection with theirs. The current QBASIC could probably be ported to the PC in less than 1 month assuming the SPL port were completed first.

Again, this would be a general CP/M 86 product, not limited to the PC. We could also sell it on the IBM Displaywriter and any other CP/M 86 machine. As CB80 sells for the Z-80 at $500 and most CP/M 86 stuff is higher, we could probably get $500 a copy retail for this without problems.

Commodity Trading Package

John Walker developed a comprehensive commodity trading package in 1980 and 1981 in QBASIC on the M9900. This package allows almost any function which would be required by a trader, and is easily expanded for custom requirements. There is no comparable package on a microcomputer. Jack Stuppin feels that there is still a market for this package, although there are several other well-entrenched competitors. I tend to agree, assuming that the package were moved to either an Apple or the IBM PC, and that superb quality user documentation were prepared (only the first chapter of the manual was written, and that was a real effort). The problems are as follows. First, there is no easy path from QBASIC to either of the ideal target machines (but

[28] That's right. To my knowledge, Ashton-Tate has never revealed the ancestry of their flagship product.

see above project). Second, anybody who works on this package must first master the concepts of securities trading. Third, it would enormously expand the market for the package if it also handled stock trading (easy, as long as a limited number of stocks were handled, and you understood the differences). I understand the package, but don't have the time to finish it. I'd like to talk about it with anybody who is interested in it.

Autodesk Work Distribution

As the potential products came into sharper focus, so did our plans to deploy our technical resources to get them done. Here's the first proposal for allocating people to projects.

Autodesk Work Distribution

by John Walker
Revision 1 — April 23, 1982

The following is a suggestion for division of work on the development of the initial Autodesk products mentioned in the "Autodesk Products" bulletin (Revision 3).

Few specific assignments have been made to Dan Drake and John Walker because both will probably be kept busy full time answering all the questions regarding the code being moved from the 9900 environment, plus running the company and acquiring hardware and getting it to work. Both will be available as required for question answering, fire fighting, and assistance.

MicroCAD

Mike Riddle
: Port META and SPL to CP/M-86. Initial conversion of SPL code, consultation with others on projects.

Greg Lutz
: General familiarisation with product, work with IBM PC, experiment with graphics display and digitiser interface to PC. Eventual, prime support and development of MicroCAD.

Keith Marcelius
: Acquisition of IBM PC, study of PC and support hardware and software we should acquire, and acquisition thereof. General MicroCAD project management, particularly monitoring of Mike Riddle's development.

Jodi Lehman
: MicroCAD manual rewrite, consulting with Greg Lutz and Keith Marcelius.

Autodesk

Kern Sibbald
: Overall project management. Design and implementation of CB80 CP/M version of product.

Keith Marcelius
: Review of product regarding user interface, survey of competitive products.

Mauri Laitinen & Duff Kurland
: Development of underlying database and storage manager for CP/M. (Likely a modification of Window's line database).

John Walker
: Consultation regarding original Autodesk design and prototype system.

Window

Mauri Laitinen
: Overall project management, review of competitive products, design and implementation of final QBASIC CP/M version.

Duff Kurland
: Development of line database module for CP/M version, consultation on product design, development of user guide.

Diff

Duff Kurland
: Overall project management, product development. Research into WordStar file format, development of "change bar" version.

Jodi Lehman
: Conversion to CP/M, documentation, and testing.

Opti-Calc

Dave Kalish & Hal Royaltey
: Familiarisation with Visi-Calc, evaluation of idea and product design, specification of product, implementation, documentation, and initial testing. This is a product of great potential that we know least about, thus the most flexibility is needed in investigating it.

Communicator

Kern Sibbald
: Complete current QBASIC version, write short guide to internals.

Peter Goldmann
: Survey of competitive products, specification of final product, conversion to CB80, development of final product, documentation.

JPLDIS

Mauri Laitinen
: Investigate availability of package, obtain a copy.

Richard Handyside
: Evaluate package, convert to CP/M, evaluate documentation and rewrite or augment as required.

QBASIC

Mike Riddle
: Conversion of META to 8086, development of floating point package, math function, edit and scan routines (part of SPL port for MicroCAD).

John Walker
: Write internal documentation on QBASIC threaded code interpretation and QP2 operation.

Dan Drake
: Write internal documentation on QBASIC I/O library design and philosophy.

Yet to be assigned—several people
: Convert QBASIC library to 8086, test compiler, upgrade documentation to IBM quality, add all CB80 features.

General research and development

The following items are not directly related to products (though some of them may be integrated into one or more of the products listed above). They are of importance to Autodesk's business.

Apples

What is an Apple? What languages exist on it? How can CP/M programs be best installed on an Apple? What percentage of Apples have Softcards? How standard is the Softcard CP/M? How are programs best gotten on Apple discs?

Jack Stuppin
: Consultation regarding Apple system experience.

Mike Riddle
: Consultation regarding Apple internal construction.

Hal Royaltey
: General product management, investigation of the Apple world.

5 1/4 inch discs

What formats are popular for 5 1/4 inch discs? Which are used by CP/M systems? Specifically, what physical hardware and controllers are used by: IBM PC, Apple/Softcard, HP-125, Xerox 820, Micropolis, Northstar? What is the best configuration for an "Octopus" machine to allow us to write discs for all these machines?

Screen driver package

Survey of existing screen manager packages. Choice or development of a general screen input package. Development of a general file driven message package.

Rudolf Künzli
: Overall project management, selection and/or development of our package for CB80 implementations.

Documentation production

How do we prepare documentation? We have an immediate need to produce "Digital Research or better quality" manuals and a near-term need to produce IBM-quality typeset manuals. We need to survey the tools for preparing copy for such manuals and the services that exist to turn this copy into camera ready copy.

Duff Kurland
: Complete and debug 9900 WORD proportional spaced Spinwriter output, edit manuals into compatible form, and print on Spinwriter.

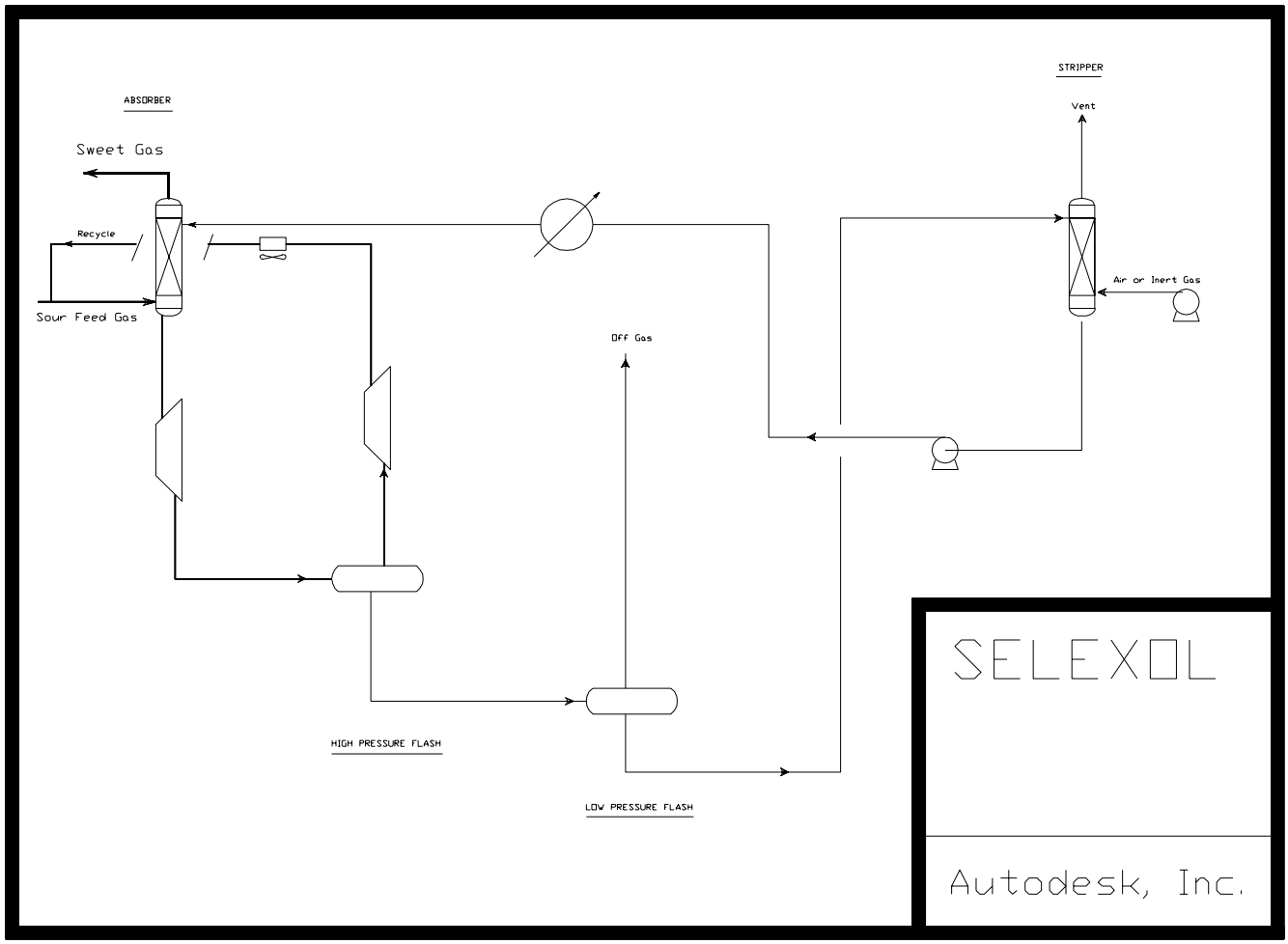
Dave Kalish
: Survey "disc to type" vendors, determine optimal formats.

Richard Handyside
: Consultation, what do we need, what should we be doing from the standpoint of a publisher.

Software manufacturers

Survey existing companies that copy discs to order. What services do they supply (format conversion, etc.), and what formats do they support? Do they put on custom labels, etc., and what must we give them?

This is one of the first drawings ever made by a customer with AutoCAD. Jamal Munshi of MOMS Computing, our first dealer and customer, drew this schematic of the Selexol chemical process. He kindly allowed us to use it as a sample drawing.



Information Letter 6

Autodesk's development efforts were officially underway when the Information Letter 6 was written in late May 1982, but nobody had yet begun to work on AutoCAD. We were still hoping to port the SPL compiler and use the original source code. And besides, AutoCAD was not seen as our flagship product. But, at last, our marketing was underway.

Autodesk, Inc.
Information Letter # 6

by John Walker
Revision 5 — May 26, 1982

This information letter is a random collection of news notes, technical information, and reports on progress on various fronts.

Marketing Progress

Since we started talking about forming the company, we've been looking for somebody to join the company with a strong marketing background and extensive knowledge of the computer field. At last our search has ended. Mike Ford, a former Vice President for Sales at Information Systems Design, has agreed to join the company on June 1, 1982. Mike will be buying into the company on the same basis as everybody else, using the option offering we're doing for the out of state participants.

Mike's marketing experience in the computer field goes back to 1956. He has worked for such industry giants as IBM, RCA, and Univac, and was instrumental in rescuing ISD in its time of peril. Since 1977, he's been running a company he formed to provide employers with employee benefits statements for their personnel. The package which does this was designed by Mike and is written in CBASIC under CP/M. Mike is also becoming a dealer for the Victor 9000 machine (made by Sirius, and sold under that name outside the US). The Victor is an extremely attractive 8088 based machine which offers twice the memory, 5 times the disc capacity, double the graphics resolution, and built in serial ports for the same price as the 2 disc IBM PC. Mike's connections with Victor not only allow us to obtain these machines for internal use at attractive prices, they give us the contacts to sell our software to Victor and Sirius. The Victor machine is an ideal host for MicroCAD, as the basic package has the graphics resolution, enough memory and disc, and the serial ports needed to do serious

work with MicroCAD.[29]

At the moment Mike is preparing marketing plans for our various products and trying to compile prospect lists and publicity channels. If you know reviewers for magazines, who to contact to get a computer store to try out a package, somebody who can get press releases run, or any such information which might be of use, please pass the information on to Mike either directly or via myself.

IBM PC

We're now the owners of a IBM Personal Computer. We bought the full-blown configuration with two discs and 64K of internal memory. We've ordered, and soon expect to receive, a "Baby Blue" which will provide 64K of additional memory plus a Z-80 processor to allow the IBM to run CP/M, and a Quadram board which will give us a serial port, time of day clock, and 64K-196K of additional memory (we're ordering 64K, and will add the chips ourself to expand it to the maximum). We've received the IBM Macro Assembler, which we will be able to run as soon as we get the requisite 96K (!) installed in the machine. We've also obtained an 8087 chip which we'll install in the machine to give it hardware floating point capability. This will both let us certify our software floating point package and let us offer the hardware floating point as an option in all the software we develop. (This will make MicroCAD immensely faster.)

At the moment the IBM is at Greg Lutz's house in the east bay, where Greg and Keith Marcelius are gaining familiarity with the machine. We have two copies of the technical manual for the machine, which we will circulate to those interested in it.

At the moment we're mostly playing with the machine and trying to figure out the assembly language. The machine's major immediate application will be to support the conversion of MicroCAD and QBASIC.

File Transfer

Because we're faced with so many different types of disc formats, we've decided to implement a universal file transfer protocol which allows us to get both text and binary files from any machine to any other given only a serial communication port. John Walker designed the protocol and implemented a 9900 driver for it.[30] Greg Lutz reviewed the protocol, fixed some flaws in it, and is now developing an IBM PC version of the program. Once that's done, we'll test the 9900/IBM link, at which time we'll be able to trust the protocol. Then we'll be able to implement it on every machine we encounter. The protocol is provably proof against data loss, duplication, and garbling, and has sufficient redundancy that it can be used on international phone lines. It's simple enough to implement in BASIC on any machine that lets BASIC drive the serial port. There are no time-critical operations that would cause trouble in a BASIC implementation.

After the 9900/IBM test, Dan Drake will put the protocol on the Apple, using Jack Stuppin's machine, and we'll have the long-awaited way to get software over to the Apple to use with the CP/M Softcard. After this is done, we'll be able to move among the 9900, CP/M, IBM PCDOS, and Apple freely.[31]

[29] And indeed it was. For the first full year of sales, AutoCAD on the Victor outsold the IBM version.

[30] This was the first crude version of **FILETRAN**, which in various incarnations serves us to this day.

[31] This never happened.

Offsite Backup

Obviously we don't want to get wiped out if somebody's house burns down. If you're developing some huge chunk of software, be sure to keep backups somewhere else. To aid this, I've set up the following scheme. Anybody who wants to back up something can simply write a disc with the name of the thing on it and the date, plus who sent it, and send it to me. I'll just keep the discs here in a special box for AI. When the box gets too full, I'll get in touch with you and see about scratching old backups for which I've received more recent copies. I'll recycle the old backups back as blank discs the next time somebody needs them.

There's no need to keep these backups very current. It's just good to know that we can't lose everything if a disaster happens.

Floppy Discs

We're now stocking 8 inch double density single sided discs and 5 1/4 inch double density single sided discs here. If you need discs for AI work, let me know and they'll be sent out UPS. We get 8" discs for $3.20 each and 5 1/4" discs at $2.63. If you can beat that, let me know.

68000 System Prospects

John Walker has been talking with a company who's developing a 68000 CAD system about getting a loaner system from them for converting MicroCAD to the 68000 (which they would then license from us).[32] If we can work such a deal, we'd be able to get a 68000 development machine in house immediately without having to spend any money. Since one of the major advantages of the 68000 is the speed and large memory that suits it for graphics, I suspect that there's more than one software-hungry vendor who might be interested in loaning a system to get a package like MicroCAD converted to it. If you see announcements of 68000 based systems that look like good prospects (e.g., have 400×400 or better graphics and cost less than $13,000 with discs included), pass on the information and we'll contact them.

Medical Software Deal

Jack Stuppin has set up a meeting between us and a company which has been developing electronic medical instruments and wants to expand into the medical office vertical market. We'll be talking with them about developing a patient records database system to run under CP/M which would optionally interface to data collected from instruments. At this point we have no details on what they want or how attractive a deal could be struck with them to do the work, so at this point this is nothing more than a lead. I'm mentioning it here because if there's somebody who has some experience in database system development or medical applications, they should be in on the meeting or at least brief somebody who's going to be there.

[32] This was Valid Logic Systems, which at the time was interested in providing a mechanical CAD capability on their electronics CAE workstation. We ended up demonstrating AutoCAD to them and their reaction was "no thanks".

Coming to Terms with the 8086

It's become clear that the plague called the 8086 architecture has sufficiently entrenched itself that it's not going to go away. For the last month or more, Mike Riddle, John Walker, Keith Marcelius, and Greg Lutz have been bashing their collective heads against it. The following is collected information on this unfortunate machine.

I think we'd be wise to diffuse our 8086 knowledge among as many people as possible. The main reference for the 8086 is a book called, imaginatively enough, *The 8086 Book* published by Osborne. This is the architecture and instruction set reference, but does not give sufficient information to write assembly code (of which, more later). However, it is the starting point to understand the machine. AI will reimburse the cost of your buying this book, which is available at computer and electronic stores.

I have never encountered a machine so hard to understand, one where the most basic decisions in designing a program are made so unnecessarily difficult, where the memory architecture seems deliberately designed to obstruct the programmer, where the instruction set seems contrived to induce the maximum confusion, and where the assembler is so bizarre and baroque that once you've decided what bits you want in memory you can't figure out how to get the assembler to put them there. But I digress.

Mike Riddle has come up with the following programming rules for the 8086. They are presented here for comments from people with 8086 experience.

- Always pop what you push. Every subroutine should leave the stack the way it found it (except for arguments and results passed on the stack).
- Assume ES=DS. This lets you use string instructions without first loading ES every time. If you change ES, put it back to DS before you return.
- Assume CLD mode set (string instructions increment SI & DI, not decrement). If you use decrement mode, restore CLD before you return.[33]
- All subroutines save and restore all registers except AX and flags (controversial, but Mike says it helps when calling subroutines in loops).[34]
- Parameter passing: Pass table and string addresses in SI and DI. Pass byte parameters in AL, word parameters in AX. Use DX:AX to pass 32 bit data. Don't use BX or CX for parameters or results; you'll need them for code that calls the subroutine. Don't use BP for anything other than stack frames. The same rules apply to results from routines.

With regard to other 8086 developments, Hal Royaltey is writing a floating point package for the beast. The floating point package will be compatible with the IEEE double precision format used by the 8087. We'll set things up so that a program can be easily (maybe automatically?) configured for hardware or software floating point. This floating point package will be used for both SPL and QBASIC programs.

John Walker has a version of QBASIC that generates 8086 assembly code. The compiler still runs on the 9900, where it will stay until META is running on the 8086. Soon we'll be loading the code onto the IBM to make sure it assembles properly, and to check out the segmentation structure of the code/library interface. Assuming

[33] Step into my parlour, said the 8086 to the naïve programmer....
[34] Utter nonsense.

that works, it's full steam ahead with QBASIC on the 8086. John Walker will be completing the compiler conversion and basic library routines, Dan Drake will be converting the I/O library, and we'll be integrating Hal Royaltey's floating point package and Mike Riddle's format independent math routines.

We'll be completing the META port on the IBM here, freeing Mike Riddle's time to concentrate on the SPL compiler and runtime library.

In developing both SPL and QBASIC, we're taking the following approach to the 8086. We want to treat the thing as if it had true large memory, even though it's deliberately set up to obstruct us in doing that. We're imposing only the constraint that the static code generated by any one compilation cannot exceed 64K (which would be an unwieldy source program anyway). Dynamically allocated strings and arrays may be anywhere in the 1MB addressing space, and linked lists will use a general segment/offset 32 bit address for pointers. Any number of modules of up to 64K each may be linked together, and runtime library size will not subtract from the maximum program size. Thus, our compilers and their generated code will be limited only by the physical memory constraints of the machine and the operating system we're running under. This is a very important competitive edge: remember that most 8086 code is translated 8080 code, and such converted code cannot easily exceed 128K (or 64K if it's messy). Our programs will have no such limit.

It's planned that an "engineering test version" of QBASIC will be running in about a week on the IBM to verify the basic memory architecture ideas that go into the above (such a test is required because the IBM assembler and linker are so confusing that whether some ideas will work cannot be determined from the manuals).

We also lack documentation of the Microsoft/IBM relocatable code format used on the 8086, although Mike Riddle suspects it's an extended version of the bitstream code used by Microsoft Fortran on the 8080 and adopted by Digital Research. Even if it is, we still don't know how the additional information for the 8086 was encoded. Does anybody know this, or have any leads to find out? We need to know to make our compilers salable, as we can't expect people to buy the IBM Macro Assembler just to assemble the code from QBASIC. I can think of lots of things I'd rather do than reverse-engineer somebody's bitstream relocatable format.

Black Hole Alert

As most of you know, Marinchip has been negotiating with Lifeboat for many months about selling MSL's 9900 software to Pertec for use on a 9900-based machine they make called the PC1000. This deal has been off and on so many times I can't even begin to recount the story. Now Pertec has officially announced the machine, including the "SB-99 CP/M Compatible Operating System from Lifeboat Associates". This is presumably Marinchip's software (unless we've been spectacularly double crossed). Nonetheless, there's no signed agreement between anybody to do the work, nor have we heard anything about this other than what we read in *InfoWorld*.[35] The reason I'm bringing this up is that if this does go through, I (John Walker) will probably disappear for a month or so into doing this conversion project for Pertec, and Dan Drake will probably be sucked in to some extent. This means that we want to get as many AI things running smoothly without our involvement as we can in case this happens. As a result, if there's something I should be doing or which you need me to do, please let me know as soon as possible so I can schedule it around this potential time sponge.

[35] One week after Pertec made this announcement, *InfoWorld* carried a second announcement saying "never mind". Pertec, a division of Triumph-Adler, heavily funded by Volkswagen, apparently came to its senses and re-thought introducing a TI9900-based business computer in 1982.

Monthly Meeting—Subchapter S Form

Remember that the first regular monthly meeting will be on Sunday, June 6, 1982 at Jack Stuppin's house in San Francisco. We haven't decided whether to go ahead with the Subchapter S election mentioned in the last Information Letter or not, but as the form requires lots of signatures, we're going to get them so we have them if we decide to do it. We should have the form at the meeting. If you can't make it, I'll see that the form is routed to you after the meeting.

Autodesk Status

Kern Sibbald has been restructuring Autodesk from the hacked-up demo version of the program written for the Computer Faire into an honest program which will run on CP/M. In the process, he had to invert the internal structure of the program because the original program heavily used recursive function calls, which aren't supported by CB80. He also installed the new general terminal driver from the CP/M Window, which allows adaptation to new terminal types simply by making entries in a file. We expect the basic system to be running on CP/M under CB80 within 30 days. At that point we can start to add the features we need to complete the system for sale.

Communicator Status

Kern Sibbald's CB80 version of Duff Kurland's TS program has been turned over to Peter Goldmann, who has successfully generated it and is now reviewing other communicator programs to choose ideas for extending the package. We will be adding autodialing, a database of systems with automatic configuration for various protocols, micro to micro file transfer, and many other features to make it the premier microcomputer communication utility.[36] It will eventually be integrated with Autodesk to add an electronic mail facility to Autodesk.

Random Bits

Does anybody know about WordStar? We need to figure out how the files work so we can fix DIFF to make change bars for WordStar files. Also, we should look at the product in general to see if we should use its conventions for control keys.

We're moving along with preparing professional looking documentation. At first, we'll be using WORD on the 9900 as our documentation tool, because we have it, we control it, and we can make it do the things we need done. We'll be installing an INDEX command and writing an INDEX postprocessor so all our manuals can be indexed. We'll install the commands we need to generate the control sequences for font selection, point size, underlining, etc., in the final output medium we use. We'll add Knuth's hyphenation algorithm from TEX, with an override ability when you see that it's botched one.

Richard Handyside told us about an outfit called "TypeShare" in Los Angeles which you dial up with your

[36] Nothing ever became of this project and product.

modem, send text with control information, and get back camera ready type. I called them up to get information, but haven't received anything yet. I hope the typesetting is faster. Other potential leads on services like this would be appreciated.

We're checking out the option of making the manuals for our stable products into hardbound books.[37] What else conveys a comparable feeling of stability and solidity? From our initial checking, it's also cheaper than the little looseleaf binders IBM uses for the manuals for the PC. Keith Marcelius is running down this option.

We've found a distributor who seems to think we're a computer store and who will sell us most mass market CP/M and IBM PC software at pretty good prices. Check first if you need something, as we should be able to get a good discount on it. We haven't ordered anything yet, so this isn't a sure thing.

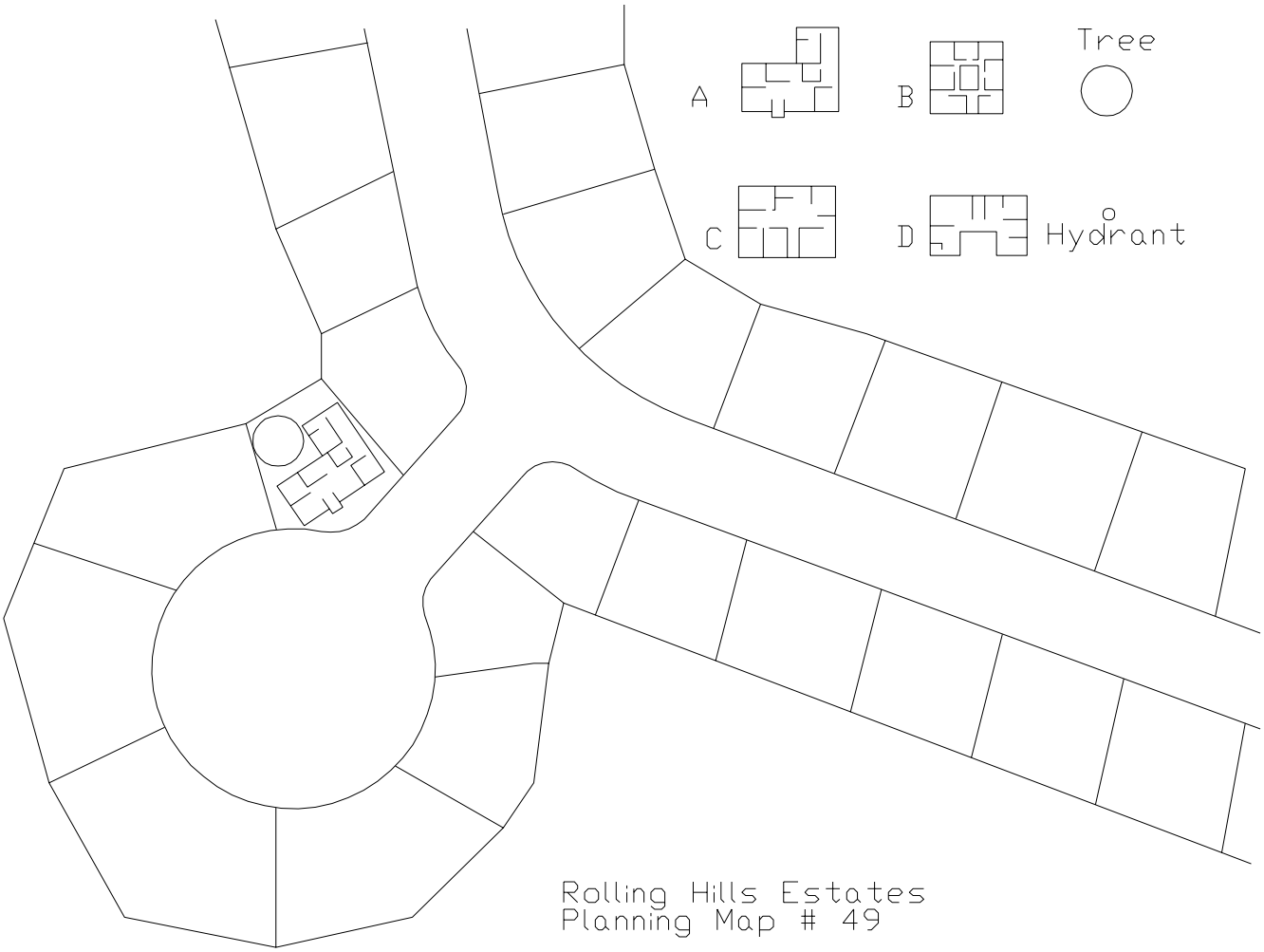
Dan Drake is currently reading up on the Apple. At the moment this is pure research, but as there are so many Apples out there, some way to get things like Autodesk on the Apple might make sense sooner or later.[38]

We're having stationery printed up. I'll be distributing it to people after it arrives, so they can use it for requesting information, etc. At the moment, the official phone number for AI is 383-2997, *but don't use it to call me, as I'm always on the other phone and get very mad when 2997 rings*.

[37] And several years later, we finally did it.

[38] This, of course, referred to the Apple II. The Lisa and Macintosh were still in the future when this was written.

Immediately before COMDEX in 1982, Sun-Flex asked us to make a demo of house placement on a subdivision plan map. Roxie and I made this drawing one horrible night in November 1982. It was while making this drawing on AutoCAD-80 that I first really became aware that much more powerful geometric facilities were needed for professional drawing. If you really want to get a flavour of 1982 AutoCAD, try making this drawing without ever using object snap, arcs other than three-point, fillet, trim, or extend.



Information Letter 7

Information Letter 7 was a general status report. It summarised an eventful month: development got underway on AutoCAD on both the 8086 host machines (in C), and on the Z-80 (using PL/I). We had given up on the old source code, and were furiously rewriting it in the new languages. It included the initial documentation of Auto Book, "the product that would not die".

Autodesk, Inc.
Information Letter # 7

by John Walker
Revision 5 — July 8, 1982

This information letter is being mailed with the minutes of the July general meeting appended to the end. Other items are included about things not brought up at the meeting.

Where We Are and Where We Are Going

A month ago I had the feeling that the company was spinning its wheels and getting nowhere. Now I think that most of the problems we had getting under weigh were normal start-up problems, which are being resolved. Definite progress is being made on all of our major products, and we can see a clear path to completion on most of them.

We're still collecting the tools, hardware and software, that people need to get the work done. If you're still waiting, be assured that you will not wait forever (or even better, help us out in getting what you need). I'll try to summarise the major project status below.

We've established a more formal structure for the monthly meetings, patterned after the original Working Paper suggestions. This form (described in the attached minutes) will minimise the "endless miasma syndrome", and allow the gist of the meeting to be condensed onto paper for those who cannot attend. Furthermore, we hope that the new format will let everybody know exactly where every project stands and what everybody is doing. At the end of each meeting, everybody should know exactly what they should be doing, and how it connects to all other work in progress.

As originally suggested, after the formal meeting we can have technical sessions on the various projects.

Autodesk Status

Kern Sibbald has completed conversion of the cleaned-up original Autodesk to CB80 under CP/M. He has prepared an internal release disc of this test version to get comments on the user facilities it offers. Each person who has CP/M capability should have already received this disc. Kern is now defining the master database that will underlie the completed system, and implementing a mockup of the database so he can convert the program to use the new database routines.

In the process of converting Autodesk to CB80, Kern segmented the program into initialisation, screen, and command overlays. This reduced the maximum size of the program to a little over 38K, so we now have a comfortable amount of space in which to work, as opposed to the 9900 version which was teetering on the brink of memory insolvency.

As soon as a real appointment calendar is installed, we will have a CP/M demo version we can begin to show to potential distributors.

QBASIC86 Progress

Implementation of QBASIC on the 8086 is progressing rapidly. Hal Royaltey and David Kalish have designed the memory model and parameter passing conventions for the object code. Hal has converted the floating point library, and is filling in the rest of the support routines prior to bulk converting the runtime library. All tools needed for this effort are in hand.

Dan Drake has made a complete audit of the differences between QBASIC and CB80 and has prepared a 10 page summary of differences and tasks required in META, QBASIC pass 1, QP2, and the runtime library to resolve the differences. He is planning to do the conversion of the compiler.

John Walker has ported META to the 8086. The port used the new C compiler we bought.[39] META was changed to generate C instead of assembly language code, and the META library was rewritten in C. As a result, META is now instantly portable to any machine which has C.[40] We may very well use C to write the second pass of QBASIC (QP2) as well (it's currently in QBASIC). If we do it all in C, the port to the 68000 will be a piece of cake as far as the compiler is concerned (since all the 68000's announced seem to have C).

The code structure which has been defined will be the first known totally general 8086 compiler implementation. There will be no limit at all on the size of a program. This should make our compiler very attractive compared to all the others that stick you with 64K data and 64K code total.

MicroCAD Status

A major change of direction in the MicroCAD project should result in completion of the conversion within the next month. Since we were able to find an excellent C compiler for the 8086, under both PCDOS/MSDOS and CP/M-86, we've decided to convert the SPL code to C rather than port SPL. Keith Marcelius surveyed the

[39] Computer Innovations C-86.

[40] But it never was. META was never used nor mentioned again after this brief note.

available C compilers and decided that the Computer Innovations compiler was the clear choice. We bought two copies of the compiler, and Greg Lutz and John Walker beat on it enough to satisfy themselves that the compiler was sound. John Walker used it to port META as noted above, and converted a set of high-precision mathematical functions to C (the Computer Innovations compiler has full IEEE single and double precision floating point, but having no math functions in the library is delivered free of SIN).[41]

The C is weak in floating point I/O, but since the compiler is supplied with complete source code for the library, and since all the relevant routines are written in C, this is easily remedied. We have found Computer Innovations to be very helpful and easy to work with, and it seems to be an outfit operating in a style very much like our own.

Greg Lutz and Dan Drake will be converting the SPL code to C. We've purchased a Houston Instruments HI-PAD digitiser which we will hook up to both the Victor and the IBM PC to test MicroCAD. We're currently looking at plotters, and are trying to see if we can work some kind of cooperative marketing deal with Houston Instruments if we use their plotters as well as their digitisers.

Our current plan for MicroCAD is to have a root segment which contains all the device-dependent parts. That segment will load the "guts" of the package which will be totally machine-independent. This has the advantage of modularising the package, making it easier to field-configure, and getting the potentially large drivers out of the address space of the package itself (it takes 40000 bytes to hold the screen bit map on the Victor).

John Walker has undertaken the task of trying to shoehorn MicroCAD onto the 8080. The effort seems worthwhile investigating as the potential market a success would open up is enormous. The effort was initiated as the result of the question "Have you ever encountered a program you couldn't make fit on any machine?".[42]

Window Progress

Mike Ford has been testing Window on his CP/M system and has found a couple of bugs which will be corrected. Duff Kurland and Mauri Laitinen have been converting the line database to 8080 code. The memory-only version is complete and currently being tested. After that's checked out, the disc stuff will be installed and we will have a product ready to go out the door (pending documentation upgrading).

For those of you who haven't used Window on CP/M, I'll mention that we've installed a completely new terminal configuration mechanism which completely eliminates the need to compile terminal drivers and link them with WINDOW. The terminal is totally described by a master terminal definition file. We plan to supply a menu-driven program which generates and updates these terminal descriptions.

Task Lists

As mentioned in the minutes of the meeting, we've requested everybody to funnel in a list of tasks before each meeting so that we can print them in the minutes and let everybody know what the others are accomplishing. The process of getting these tasks in and concentrated worked so poorly this time that I don't think it makes sense trying to summarise them here—it would likely create more confusion than it would dispel. Please keep

[41] These math functions are still used in AutoCAD.

[42] He made it fit. Two years later the product was discontinued after selling about 150 copies.

this in mind in the future—each monthly letter from now on should contain detailed task lists. I'd like to work out a way (maybe via MJK) that I can prepare the task summaries without having to retype pages of information.

Conference System Notes

Most people in the company are now using the MJK teleconference system to interchange messages. As requested, I've added the conference system user names to the "Names and Addresses" directory at the end of this letter (after the phone numbers).

The conference system has been afflicted by the recent times of tribulation in TYMNET. TYMNET has been installing new software, and we've been through yoyo reliability, double spaced input lines, character delete that comes and goes, etc. All we can do is put up with it. The problems are in TYMNET, not the conference system.

Please note that charges for the conference system, including TYMNET connect time (the largest component of the cost), drop by more than 50% in non-prime time, that is, 18:00 to 07:00 Pacific local time (the time that's displayed when you log on). The reduced charges also apply on Saturdays, Sundays, and holidays all day. Please help us save money by using the system when it's cheap. Note also that MJK's preventive maintenance remains from 17:00–17:30 Pacific time, so if you try to log on at that time you'll get "Can't initiate new sessions now".

To further reduce charges on the system, I've reversed a change made to the system some time back and made the system store messages for people in files keyed with their names. This means you can check whether there are any messages for you without calling the CONFRX program and incurring the charges to load and execute it. At the point you're about to type RUN CONFRX, type LISTF instead. You'll see a file directory listing. If you see a file with your user name preceded by "MF", then you have messages. If there's no such file, and you don't want to send any messages, you can type BYE immediately and log off. Thus, if you user name were GONZO, you would look for a file "MFGONZO" in the file directory.

Organization Details

The Subchapter S alternative we considered before turned out not to be possible after all, because one of the stockholders (MSL) was a corporation, not a person. This disqualifies AI from Subchapter S. As a result, we'll go ahead soon with the option plan to bring in the rest of the people. We should have done this already, but the press of work kept us from getting to it.

C

C is shaping up to be an important language in AI's plans. It looks like the language of choice for the 8086 based on the Computer Innovations compiler, and of course it is the workhorse on any of the Unix ports to the 68000 or elsewhere.

If you don't know C, it would be a good idea to pick up a book and start reading up on it. The reference is *The C Programming Language* by Brian Kernighan and Dennis Ritchie, Prentice-Hall, 1978 ISBN 0-13-110163-3. AI will reimburse the cost of your buying this book.

The Computer Innovations C is a full, unrestricted, implementation of the language as described in the book. If anybody knows of a similar 8080 C with good code and an attractive runtime licensing deal, please let me know.

I'm currently exploring the option of converting WINDOW to C for the 8086.[43]

Random Bits

Utterly out of the blue, Marinchip has completed the most spectacular sales month in its history. In June we sold more than our total sales for 1978 and 1979 combined. If sales were to continue at the present pace, Marinchip would be shipping at an annualised rate of $850,000 (neglecting for the moment the little detail that John Walker would disintegrate in the process). This is being mentioned because if you've been waiting for John Walker to do something for you, you'll probably have to wait a bit longer. As a result, we've tried to further decentralise the communications in AI—a lot of information was passed through John Walker simply because that was easy. But it won't work at the moment. We expect the June results to be a one-time blip since most of the sales were unexpected one-shot sales rather than dealer or OEM business. However, July looks like a barnburner as well.

John Walker has been playing around with a new product idea called Auto Book. A M9900/QBASIC test program has been developed to explore ideas. If the product looks worthwhile, we can consider it as an innovative way to distribute manuals for our products, to offer an impressive help facility, and as a product in its own right. I'm including the working paper on the product with this mailing for your review and comments. Richard Handyside and Jodi Lehman have copies of the program and are currently evaluating it and making suggestions.

On the 7th and 8th of August there will be an Autodesk/Marinchip dealers meeting in London. Rudolf Künzli, Richard Handyside, and Peter Goldmann will be there. If you have items that would make sense to bring up at the meeting, please try to get them to the people involved before the meeting.[44]

[43] Kern Sibbald eventually undertook this task. Thus was born Kern's Editor.

[44] After that meeting, I recall sitting in the office in London and getting the **TEXT** command and block insertion with scale and rotation working in AutoCAD for the first time.

July General Meeting Minutes

During this period, monthly weekend get-togethers comprised almost all of the face-to-face contact between the people involved in the company. Dan Drake's minutes of the July, 1982 meeting were mailed with Information Letter 7.

July General Meeting

by Dan Drake

The July general meeting was held on Saturday, June 26, 1982, at Jack Stuppin's house. (For those who missed it or have forgotten, the algorithm for computing meeting dates is at the end of the minutes.)

The meeting was called to order at 1:10. Present were Dan Drake, Mike Ford, Dave Kalish, Greg Lutz, Keith Marcelius, Kern Sibbald, Jack Stuppin, and John Walker.

Marketing

Mike Ford discussed his work on marketing and the questions that need answering before we go much farther. He has got two Victor 9000's on loan (one from Sirius) for work on QBASIC and MicroCad, and has scored an MSDOS with assembler and linker. Also Pascal, for which we need another 128K memory. We have given a demo for Hal Elgie, a consultant for Sirius who was highly impressed, especially with Autodesk.

Mike's Victor dealership seems to have opened some doors. He could probably get a dealership for other machines if it would be useful.

There was considerable discussion of the terms on which we want to sell the programs. Our main options in dealing with Victor, and probably anyone else, are these:

1. Victor buys the source for a flat fee, though not necessarily with exclusive rights. Victor does all support.
2. We provide a program in object form. Victor promotes it as its own and pays us royalties, probably a fixed amount per copy. If Victor is serious, we should be able to get a substantial advance when we close the deal.
3. We keep it as our own, but they publicize it to dealers and the public as one of the good things on their system.

(Many numbers were bandied about in the discussion. They are not in these minutes, because they would give a false air of precision and because widely distributed pieces of paper tend to pass before unauthorized eyes in spite of all precautions. Call us to talk about numbers if you like.)

There was general agreement with John Walker's opinion that a source buyout might be all right for a limited product like QBASIC-86, but not for MicroCad. The potential market for MicroCad is unexplored; it could be enormous, and no one would pay us enough to compensate for it. Victor could sell it under approach (2), putting their name on it if they want, but our name should at least appear on the disc and in the manual.

There was serious discussion of the right list price for MicroCad. The consensus was that the present price on the M9900[45] is probably too low.

QBASIC-86 would also be best sold on a royalty basis, though a buyout is conceivable. It should be easy to sell on the basis that it's a markedly superior language to CB-80, provided that we get it done well before Digital Research is ready.

Window is a product that would be a natural for Digital Research, which offers no usable editor for programmers; but we haven't managed to make a useful contact. Kern expressed concern that we shouldn't let it get completely out of our hands, because it's much better than anything else on the market. John pointed out the difficulty of trying to sell it ourselves, competing directly with VEDIT, which is becoming entrenched and has a large advertising budget.

We need to approach Corvus, Fortune, and anyone else who has a 68000; we should be able to get development machines from them. Also, the NSC 16032 is now approaching reality, and its speed makes it very attractive for MicroCad.

Financial report

John Walker presented a financial report. Proceeds of the sale of stock and options were $59,030. Expenditures have been as follows:

IBM PC with printer, memory expansion, etc.	6,317
Sierra Z-80 boards (CP/M for M9900 users)	1,804
Stationery, copying, etc.	324
Supplies	337
Printing for Computer Faire	820
Legal fees	3,949

Income: $115 interest from Capital Preservation Fund.

We currently have $45,592 in liquid assets, almost entirely in Capital Preservation Fund. (Yes, Virginia, there is a round-off error in the totals.)

Progress reports

Each of the people present reported on what he has been doing:

- Current commitment of time per week.
- What projects he has been working on, with what effect.
- What he's now working on.
- What obstacles are in the way.

[45] Namely, $1,000. We ended up introducing the Z-80 and MS-DOS versions of AutoCAD at $1,000 anyway.

Everyone was also asked to submit by Wednesday a list of tasks that he'll be working on over the next month or two, on a fairly detailed level. Submission can be either to John Walker or to the project managers concerned, who will forward the lists.

CPM Charts

Dan Drake has been playing with a CPM program called MILESTONE, which runs under CP/M. There are now charts for the four main projects: Autodesk, MicroCad, QBASIC, and Window. Copies were handed out to everyone at the meeting; people who weren't there should request any that they want to see, so that we don't waste airmail postage on many charts that no one wants.

The time estimates in these charts are not in any sense imposed deadlines; they started as moderately optimistic guesses, intended to avoid the most obvious pitfalls in the critical paths. Anyone who finds himself on an unreasonable schedule can submit better estimates for the tasks that he's involved in. Many of the guesses were corrected at the meeting.

Any project leaders who don't see the CPM charts as a waste of time will probably want to maintain their own. We can get copies of MILESTONE for anyone who has CP/M (80) capability. It's worth seeing MILESTONE run just to see a really well done menu-driven program.

Various business

We now have letterheads and envelopes. Business cards for everyone will be available soon, probably with the default company title of Product Development Manager.

Dave Kalish has found a publisher who takes formatted ASCII text by phone and does typesetting and printing. We need to meet with him on technical details and prices. John Walker has a list of several publishers who accept some sort of floppy disc input (compiled from much-appreciated information from Richard Handyside and Peter Goldmann).

There was a discussion of the process of bringing new people into the company. The consensus was that anyone brought in on the same basis as the founders would require unanimous consent in some form. Assuming that the proposal had already been discussed, a final decision would not have to wait for a monthly meeting: management could poll everyone (possibly through the conference system on MJK) and proceed within a couple of days if there were no negative votes.

Future agendas

For future meetings, as for this one, the corporate secretary will prepare an agenda and attempt to protect the meeting from creeping formlessness. If there's something that needs to be on the agenda, please send a message to Dan Drake or John Walker a couple of days in advance, preferably on MJK.

The agenda will include progress reports of the same sort as were given at this meeting. Reports should be well under five minutes long; anything needing more discussion will be taken up later in the meeting. With these and the written list of tasks we'll be able to keep track of what's getting done and what's slipping.

The task lists and time commitments will be published regularly.

Here is the algorithm and schedule for monthly meetings:[46]

- The meeting for an odd-numbered month is on a Saturday; for an even month, on a Sunday.
- The meeting is normally on the first <Saturday or Sunday> of the month.
- If that day falls on a holiday weekend, the meeting is held a week early.

The schedule for the rest of the year, therefore (excluding the Annual Meeting) is as follows:

Sunday, August 1
Saturday, August 28
Sunday, October 3
Saturday, November 6
Sunday, December 5

[46]This is the famous “Sibbald Algorithm”.

Auto Book Notes

I also included the original notes describing Auto Book in the Information Letter 7. Maybe someday we'll finish that product.

Auto Book Notes

by John Walker
Revision 6 — July 8, 1982
(Special IL7 Version)

Auto Book is an idea for an automated document retrieval and examination system. What exists today is a prototype intended to play around with the concepts and try out bright ideas without a lot of effort. It is implemented in QBASIC on the M9900, and no effort has been expended to make it transportable. I'd rather have the convenience of trying out ideas readily than always trying to maintain compatibility.

The Concept

Now that everybody has a computer, everybody will naturally write, edit, and print documents on the computer. As a result, more and more documents accumulate in machine readable form. Little has been done toward letting users *read* documents once they are written. If we have a computer between us and the document, we should be able to take more advantage of it than just having it print a hard copy that we later read.

Why shouldn't we be able to:

- Ask questions about the document?
- Read sections based on content?
- Move between documents as they reference one another?
- Ask for explanations of terms the document uses?

The idea I'm exploring is "Computer Assisted Reading". It's a field I've seen little done with, and it's a far more universal need than even, say, VisiCalc. I'm calling the program "Auto Book" because it implements an automatic intelligent book. It also might be called the "Reader's Workbench".

`READ` - Reading an Auto Book

To get started, you should edit the file "ADI.CFG" on the Auto Book disc and change the "TERM" statement to specify your terminal type. The terminal type specified must be one listed in a "*" line in the file "WINDOW.TRM" on the disc. This is exactly like the CP/M Window configuration, and you may use the notes for that product to aid in configuring new terminals.

Put the Auto Book disc in Drive 1. Call the read utility with the command:

READ

When you are asked for a document name, answer:

USC

If you're about to ask me about the plus signs, it's because the terminal definition defaults to plus signs for "flagged" lines. They're reverse video on my terminal. If you have a special display mode, make the entries in WINDOW.TRM to use it on your terminal.

READ Commands

You will see a menu of commands. **READ** works in two major ways, by locating sections of a document by content, and by working with marks in the document set by the user while viewing it (just like attaching paper clips to pages, inserting bookmarks, or dog-earing pages). There is a list of all "referenced" sections known to **READ** at any time. Keep these ideas in mind as you read the explanations of the commands (it is a good idea to play around with the program as you read these sections).

Add word to references. You're asked for a word. All sections in which that word appears are added to the list of referenced sections. (That is, in the logical sense, they are OR'ed.)

And word with references. You're asked for a word. All sections in which that word appears, and which have been previously marked as being referenced are marked as referenced. Other previously marked references are cleared.

Subtract word from references. You're asked for a word. All sections in which that word appears are removed from the reference list (if present).

Clear references. All references are cleared from the reference list.

List references. The titles of all sections referenced are listed. If more than a screen full of titles are referenced, the user can type "M" to see the next screen-load. "C" returns to the command menu.

Show referenced text. The text from the document is displayed for referenced sections, starting with the first referenced. If the text is more than a screen full, typing "M" will show the next screen-load. Typing "N" shows the next reference, if any. Typing "C" gets back to the command menu. "+" advances to the next section in the document (referenced or not), and "–" backs up to the previous section. "S" sets a mark on the section, and "U" unsets (i.e., clears) the mark.

List section titles. All section titles are displayed. "M" gets the next screen full, and "C" gets back to the command menu.

Select marked sections. All sections which have been marked by the user (with the "S" key while viewing the text) are added to the list of referenced sections. To view just the marked sections, clear the reference list, then select marked sections.

New document. The user is asked for a new document name, and viewing of that document is begun.

Help facility.

When viewing a screen other than the main command menu, pressing any illegal key ("?" is always illegal), will replace the display at the top of the screen with a list of the meaning of all the currently valid response keys. Entering a proper response will turn the help display back off.

SCAN - The preprocessor

Auto Book consists of a document processor which reads the formatted text generated by a word processing program. The preprocessor scans the document, and based either on information encoded in the document, or by user-selected heuristic rules, identifies logical sections of the document and assigns them names. It prepares a rapid-access file of the document text, and creates a file containing encoded references to words in the document with pointers to the sections of text in which each word appears. The preprocessor may also perform compression of the text, and encode it against access by programs other than the Auto Book retrieval program. Neither of these functions are currently implemented.

The preprocessor contains an algorithm called the "rooter" which extracts the root of words with prefixes and suffixes. This algorithm must be carefully defined, and will vary for each natural language supported. References are stored by the root of the words, so that asking for references to "test" will find references to "test", "tested", "tests", "tester", "retest", etc. This is not currently implemented.

Once a document has been "compiled" by the preprocessor, it may be read with the "**READ**" utility.

The preprocessor is invoked by calling the **SCAN** program. It presents a menu allowing only the options of preprocessing a document or exiting. Before calling **SCAN**, you should have put the **WORD** formatted output of the document into a file with a .TXT type. You should also create a file with the same name and a .RAT type with size about one sector per line of text in the .TXT file. You should also create a .REF file. There's no easy way to estimate the .REF file size, so make a huge file initially. **SCAN** will tell you how much it used after it's done. You should have a **TEMP1$** file on Drive 1 (MDEX) before calling **SCAN**.

Once you tell **SCAN** you want to process a document, all you have to do is enter the "root name" (less the .TXT) of the document, and **SCAN** will do the rest.

SCAN knows about various commands embedded in the document. These commands will be removed from the files created by **SCAN**. All commands are flagged with a plus sign (+) in column 1. Note that these commands are entered as text with **WORD**, and that care must be taken to insure that **WORD** will not format them into the

middle of another line. See the file "USC.WRD" for an example of how the **SCAN** commands can be inserted in a document.

SCAN Commands

The following commands are recognised by **SCAN**.

+TITLE *text*

The *text* is saved as the document title. The title is always displayed while the document is being read.

+COPYRIGHT *text*

The *text*, which should be of the form "1980 Mud Slingers International" will be displayed as a copyright notice when the document is being viewed. This may, in the future, be used to control reproduction of an encrypted document.

+*number text*

If a single digit number from 1 to 9 follows the plus sign, this specifies a section break in the document. Up to 9 levels of sections are allowed. The title for each section is the concatenation of all sections with numbers less than or equal to the last section number which appeared.

Important: If no *+number* item is used, **SCAN** will break the document into paragraphs separated by blank spaces. Each will be assigned a paragraph number. This allows unencoded documents to be processed reasonably.

Trying SCAN

To see the process involved in running **SCAN** at work, look at the original document text "USC.WRD" on the documents disc. This text is processed by **WORD** to form the text file "USC.TXT". When **SCAN** is run over this document, the files "USC.RAT" and "USC.REF" are generated. These files are then accessed by **READ**.

Ideas for the future

The list of commands in **READ** and the whole idea of the command menu is distasteful. I think maybe a simple command language or some form of directed prompting would be more in order. The current set of commands evolved largely out of a desire to test the various sections of the program in an orthogonal manner. I'm sure a more elegant set of ideas should underlie the commands.

You should be able to do a lot more with marks in the text. They should be saved when you sign off or view another document, and you should be able to clear them, easily display them, automatically set marks for all referenced sections, etc., etc.

There should be a +ALIAS command in **SCAN**. Sometimes you want a section to be selected when a word not used in it is referenced. For example, the "necessary and proper" clause in the U.S. Constitution might have a +ALIAS ELASTIC before it, as it is often known as the "elastic clause". It could be found by its common

name, even if the user didn't know what it said.

There's another aspect to aliases. You might want to have an alternate word or words indexed every time a given word is used. This would let relevant sections be retrieved regardless of which synonym were used. For example: +ALIAS BONNEY=BILLY THE KID . would index "Billy the Kid" every time the desperado's real name were used in the text.

Inter-document references: A complete system should let you file all your documents and move freely between them. **SCAN** could implement this with a +KEYWORDS statement listing keywords in the document for the global dictionary. One could start at the global level and get a list of all documents with selected keywords, then move on to read them. References between documents would be handled by a +SEE statement. One might, for example, in a manual about the text editor, insert the statement:

+SEE SYSTEM REBOOT,LOAD,CREATE,FILE,DELETE,...

where the user would be given a reference to the manual "SYSTEM" when one of the listed words were asked for.

SCAN should also have a command called +EXPLAINS. Before a section, one should be able to insert a statement like:

+EXPLAINS CHANGE,ALTER

and have it flagged as the section which explains those terms. Then the user reading the document could ask for the explanation of a term (rather than just the references) and get the section providing the most basic definition of the term.

When you're reading a real document, you can make marginal notes. **READ** should allow this too. The master copy of the document remains unchanged, but a user can "annotate" any section by typing in text which gets saved in a special notes file belonging to that user. When the section of the document is viewed, the user can see that he's made notes, review the notes, and edit them as desired.

The format **SCAN** stores the text in is wasteful of space, and results in each document being stored as two files (.RAT and .REF). This is because I was lazy. Fixing it wouldn't contribute anything to evaluation of the product idea. In a production system, one file should contain all information for a document, and the document text should be compressed using the "polygram compression" algorithm used in SPELL. Also, a simple encryption should be done to protect documents from being ripped off by the honest and naive. Whether the user is allowed to make a hard copy or store decoded text in a file would be controlled by a flag on the copyright statement in the document. Compression is very important because the value of this system depends on how many documents you can keep on-line.

There should be a way in **READ** to locate text by section title as well as by word references. I'd suggest a command which lets you specify words from the section title. It scans the section titles and looks for a section title containing all the words you used in your specification (regardless of order). If more than 1 were selected, you could look at them and choose the right one.

More general facilities should be available for moving around in the text when looking at text with **READ**. You should be able to:

- Go to next reference (in already).

- Go to previous reference.
- Go to next marked section.
- Go to previous marked section.
- Show previous page of long section.
- Go to next reference of specified word.
- Go based on section title.

These facilities are why I think that a "command line" at the end might be better then the menu/view mode presently installed.

Also, should we encode the hierarchical structure of the document? We know the levels based on the encoding given to **SCAN**. We might want to say, "Go to the next chapter", or such.

One of the most complicated design tasks is the "rooter". I think the guts of the TEX hyphenation algorithm are a good start. We want to index the original word, then add the derivative forms, flagged as such. Then we can retrieve based on the exact form, or any derivative form.

I don't expect most people to make as much effort encoding a document as we might make in indexing manuals for distribution with this thing. Thus, **SCAN** should be far more intelligent in breaking up documents into sections based on heuristic rules. We'll need to learn what information there may be in a WordStar file, for example, which would help in this task. The utopian idea is that once any document is written (letters, etc.), in an office, the original text is archived, and the formatted text is run through **SCAN** and saved on-line. Anybody who refers to it does so with **READ**. This makes references more productive, saves disc space, aids in building a master document library, and allows readers to make annotations without either changing the original or copying it.

Don't be upset by how slowly **SCAN** runs. I used a stupid merge algorithm in sorting word references. It should be able to be speeded up to run faster than **WORD**. For evaluation, it serves.

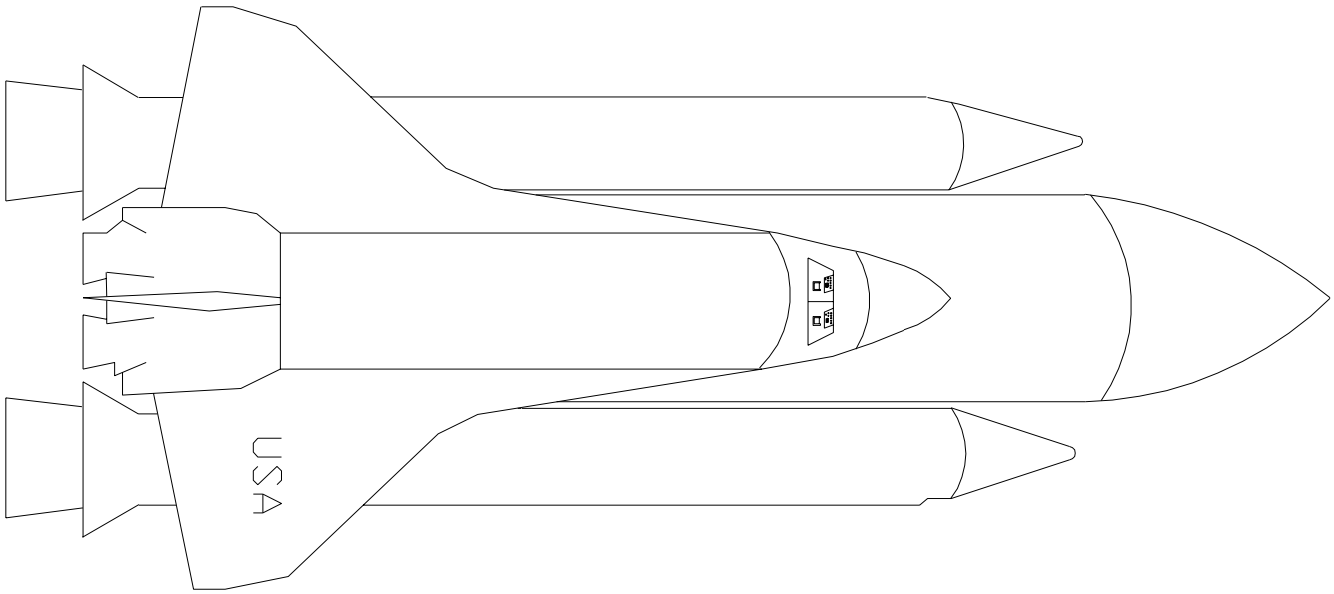
Yes, **READ** is awfully fast, isn't it? The sneaky way it looks up indexed words remains fast even with very large documents.

You can also use **READ** to aid in access to paper documents. To do this, just make **WORD** crank out a:

+1 Page *number*

item in the HEADING macro. Then all sections will be flagged with the page number.

This drawing was, to my knowledge, the first actual drawing ever done with AutoCAD (other than scribbles made while testing the program). I initially drew it on AutoCAD-80 before text was even working, then added detail as parts of the package were implemented. The drawing was made by taping the cover of a Time *magazine issue from late 1981 featuring the shuttle to a HI-Pad digitiser and tracing the drawing. The picture in the magazine wasn't precisely a face-on view; that's why the drawing is slightly asymmetrical. This drawing was also the first BLOCK ever used with the INSERT command, and the first drawing ever to be plotted with AutoCAD (on a Houston Instrument DMP-8 plotter). The CHANGE command was initially implemented to help clean up the raw digitised coordinates in this drawing.*



Information Letter 8

The press of working flat-out on AutoCAD and trying to run Marinchip at the same time began to tell on Dan Drake and myself. This brief note was the last Information Letter for well over a year. As the pace of activity in the company accelerated, verbal communications over the telephone, via the MJK teleconferencing system, and at the monthly meetings supplanted written summaries.

Autodesk, Inc.
Information Letter # 8

by John Walker
Revision 2 — July 25, 1982

This information letter is to suggest a new participant in the company, Stephanie Nydell. Her participation letter is attached, which details her background and interests.

Those of us who are ISD old-timers don't have to be told that Stephanie would be an excellent addition to the company. She would be able to immediately get to work on documentation production, and having written manuals for a competing CAD system, is eminently qualified to tear into the MicroCAD manual. Also, she can become the nucleus of our customer support operation, having experience in that at both Nicolet and at Information Unlimited Software.

All the rest of the details are in the letter, so read it. Stephanie will be at the August 1 meeting, so you can talk to her and ask any questions you might have at that time. Assuming there's no objection, we can bring her into the company shortly after the meeting in the long-delayed option issue.[47]

Other Items

Everything else is moving ahead rapidly. MicroCAD-80 (the 8080 version in PL/I) is pretty much running now. The 8086 version has now been completely translated to Computer Innovations C, and Dan and Greg are compiling away and beginning to stack modules into a complete program. Greg has the graphics driver for the IBM PC finished. We've received the digitiser, and it's now running with MicroCAD-80. We finally received our copies of CB80, and they're on their way to the people working with CB80. We got a WordStar,

[47] For various and assorted reasons, she did not end up joining the company.

MailMerge, and SpellStar at about 80% off list for documentation use and feature evaluation. Duff and Mauri have the memory-only version of the Window line database working, and are moving on to the backing file pager. Kern has the real appointment calendar in Autodesk, and is designing the database interface. A test version of Communicator has been put together by Peter Goldmann, and Rudolf Künzli will be completing a new version of the screen driver package shortly. We now have the information we needed on the 8086 relocatable format. The QBASIC86 project now has no obvious stumbling blocks, and is moving ahead. We've found two typesetting services so far which accept ASCII text, and we're comparing them and others to choose a way to produce manuals.

See you at the meeting.

August 1982 Meeting

As the company moved from organisation to operation, communication which had previously been conducted through the Information Letters moved more and more to verbal communication at the monthly meetings of participants, as well as ongoing telephone communication among project teams and messages on our teleconferencing system ("MJK"). Dan Drake inaugurated the publication of summaries of the monthly meetings with this document.

This was particularly eventful time. The decision to scrap the QBASIC version of Autodesk and begin rewriting it in PL/I was made. Development of the initial version of AutoCAD was coming to a close; we decided that versions would be developed for the Scion Microangelo, IBM PC, and Victor 9000, and that the initial price would be $1,000.

August General Meeting

by Dan Drake

The August general meeting was held on Sunday, August 1, 1982, at Jack Stuppin's house.

The meeting was called to order at 1:10. All company people in the Bay Area were present: Dan Drake, Mike Ford, Dave Kalish, Duff Kurland, Mauri Laitinen, Greg Lutz, Keith Marcelius, Hal Royaltey, Kern Sibbald, Jack Stuppin, and John Walker.

Changes in Participation

Stephanie Nydell, whose interest in joining the company was discussed in the previous information letter, was introduced to the meeting. There was unanimous agreement on bringing her in as a founder, subject to approval by the people out of state.

There was a brief discussion of the mechanics of buying back the shares of Jodi Lehman, who has decided to leave the company. It is possible that the company can buy them back, or a new participant could buy them. It appears that there will be no difficulty in working out the details with legal counsel.

Minutes and Financial Report

There was no dissent from the published minutes of the July meeting. John Walker presented the financial report. The balance of funds last month was $45,592. Expenditures have been as follows:

Supplies	85
Softcard for Apple	382
Digitizer for MicroCad project	665
C compilers	42
Apple documentation	?
Total	1,671
Income: Interest from CPF	325
Current Assets:	44,426

Marinchip Systems, Ltd has for various reasons bought items for ADI worth about $1,542. ADI has also agreed to buy hardware from Duff Kurland in the amount of $5,115.[48] This makes:

Current Liabilities: 6,657

Planned expenditures include compilers (C, PL/I, CB80), business supplies such as invoices, and computer equipment, as yet unspecified, for the use of Mauri Laitinen.

Progress Reports

Everyone delivered a report of 2–3 minutes covering:

1. How much time are you currently putting in?
2. What have you been doing and what accomplished?
3. What are you doing next?
4. What's currently standing in your way?

Progress on various projects will be discussed later. An important feature of the reports was that there were almost no reports of obstacles to getting the work done.

[48]This was the Marinchip computer, Morrow hard disc, and NEC Spinwriter which was, almost three years later, to become a cause célèbre during the public offering as a "promoter transaction".

Random Business

There are some CP/M utilities available to anyone who wants to use them within the company. Some are not ADI property and should not be distributed outside.

DELETE Does what the DEL command in MDEX does with an ambiguous filename, bringing up each selected file for a choice of whether to delete it. It will also find all illegal filenames, which Microsoft programs like to create.

DOC A primitive subset of WORD.

FDUMP A file dump that shows file contents in hex and ASCII.

Project Discussions

Marketing

We have the current IBM software distribution agreement. Anyone who is curious about their standards can get a copy.

In addition to Mike Ford's connection with Victor, we have good contacts at Apple, Hewlett-Packard, Onyx, Corvus, and Timex (which sells the Sinclair). Most of these are just waiting for us to have products to demonstrate.

Dan Drake brought up a possible guerrilla project: an existing Marinchip program that computes all known tax deadlines. The complexities of tax payments are such that the Wall Street Journal ran a flowchart on Federal withholding tax deposits alone. A program that handles these details ought to be worth $50 to any business in California. (It would be a major project to configure it for other states.) ,

The consensus was that we should convert the program to CP/M and put it on the Apple Softcard. We should get an opinion from accountants, though they have a conflict of interest in that they get high fees for providing the same information to their customers. The program will have to be sold with the understanding that it will be obsolete in a while; people will have to get updates when the rules change. It would be appropriate to let someone else distribute it, since it's off our main line. A major problem is putting on a good enough disclaimer to keep from being sued into nonexistence if we, the customer, or the tax people misunderstand the rules.

John Walker described Micropro's anti-theft provision: their manuals say, in places that make the statement hard to get rid of, that if your copy of Wordstar doesn't have such and such a sticker on the label, you have an illegal copy. Just send it to us and we'll give you a legal one! Of course, they can find the buried serial number on the disk, so they know who the fink is. This seems the best protection scheme in the business, and well worth emulating.[49]

The most animated discussion of the day was on marketing MicroCad, which is very close to being a reality. The plans agreed on are summarized as part of the MicroCad project discussion.

[49]This was the rationale behind the notorious "metal labels" we used on our discs in the early days. We never did make the offer to redeem pirated copies, largely through oversight.

Autodesk

There is now a functioning calendar, and the data base is being redesigned. After a short discussion of features and problems, John Walker raised what seem to be fundamental problems in the project. What follows is approximately a logical, not a chronological, summary of the discussion.

The original quick and dirty implementation in CBASIC is now running on CP/M systems in CB80, and Kern Sibbald has put hundreds of hours into conversion and making it run well, but we seem no nearer to a salable product than we were at the Computer Faire. It's still of no practical use, being unreliable and terribly slow.

There was disagreement on the extent of the non-progress, but a consensus that things were not moving nearly fast enough.

The version at the Faire looked like an outstanding product, but it glossed over many crucial technical problems which must be resolved before anything is sold. These don't have obvious answers, and the attempt to fit the answers in as we go along has given us a program as big as MicroCad that does less and runs slowly.

One technical problem may be the wrong choice of language. CB80 lacks the right I/O facilities and requires fairly massive assembly language interface routines, which are especially clumsy because of the lack of data structures. In C or PL/I the problem would disappear. But that conversion would take some time.

Perhaps the existence of the prototype has fooled us into thinking that we could work that into a product, when actually a full re-design is needed.

If we scrapped the project entirely, we would have plenty of things to do with the manpower released, but no one wanted to do that. On the other hand, if we continue the project, we need more people involved in it.

A consensus was developed along these lines: The program will be rewritten in a better language. At present, if we want to sell to half million existing 8080 systems, that means PL/I; by the time the program is ready there should be a PL/I available for the 8086. Also, when the screen handler has been done in PL/I, we'll have gone a long way towards a PL/I version of Window.

The thing we are to produce is a user-friendly card box system with multiple boxes holding cards of unlimited size. Once it's done and on the market, we can work on further releases with added features. This first implementation may get us into some decisions that we'll regret when we start adding features, but we have to get a working product out the door in a finite time.

Dave Kalish will work with Kern Sibbald on the database design and on user features. Duff Kurland and Mauri Laitinen will work on the screen handler. As a crash project the thing could be done in a month or two; since we don't have people working full time, we must be resigned to its taking longer.

There was general agreement that we needed to know more about competing products, including MBA. No one actually said that he'd do such an investigation.

JPLDIS

We now have data on availability. The price quotes range from zero to $3,450. We are supposedly getting a free copy in 6–8 weeks, but no one is betting on it.

MICROCAD

Both the PL/I version on the 8080 and the C version on the IBM PC are now converted and running, needing some amount of work before release. The Victor version will be running soon. Without any optimization or division into overlays, the 8080 version has 4K bytes of memory left over, and runs as fast as the 9900 version.

The discussion of marketing went very roughly as follows.

John Owens has sold perhaps 20 copies of the 9900 version through his ads in Byte, and has had lots of people ask why it isn't on the 8080. He might sell as many as a hundred in a short time if we turn him loose.

There are some large markets for an 8080 MicroCad. Scion is selling enough Microangelos to support a color ad in the front of Byte every month, though there's very little software available. A version using the Microangelo with a light pen would be a completely standard product that would run entirely on Scion's graphic equipment and might be something they would want to market. There are other markets further along, like Apples and the incredibly cheap Sinclair. Again, it might be easily converted to Univac PL/I, running on an 1100 with Tektronix graphics.[50]

On the other hand, since we could ship out copies and sell a few almost immediately, there are marketing decisions that must be settled immediately. What do we do about Beta testing? Do we sell a pre-release? How do we price it? What do we know of the reactions of live users of the 9900 version? The only people in the company who have spent any time using it are computer fanatics who know the internals and are not well qualified to judge user interfacing for a large market.

It's clear that getting some copies into the field and getting information back from end users will be invaluable in developing the product.[51] Getting some actual money into the company is also important.

Fears of getting a hostile review on an early release were dissipated by the observation that it's extremely hard to get reviews when you want them. A product that's selling at most a few hundred copies is in little danger of any reviews.

The consensus reached was this: The 8080 version will get a driver for whatever plotter John Owens currently likes to sell.[52] Stephanie Nydell, who has done documentation and support for larger graphic systems in the electronics business, will write up suggestions for making the user interface more friendly to completely unsophisticated users. She will also make minimal changes in the current manual to make a preliminary version that we can send out with the early release. We'll tell Owens that the product is coming, and he can sell it to his 8080 prospects. On Wednesday, August 25, we want to send this version out.[53]

There will be a Victor 9000 version for Mike Ford to demonstrate by the end of September.

The initial list price will be the same $1,000 as for the current 9900 Interact.[54]

Once we have this product out the door, we demonstrate to Hewlett-Packard and pursue our other contacts. Work will start then on a new manual and will proceed as we exercise the program and get feedback from users.

[50]This was the first and, thankfully, last mention of this idea.
[51]Thus was born the idea of "let the user wish list design the product".
[52]Today this is called a "market driven peripheral support strategy".
[53]First committed ship date for AutoCAD or, for that matter, any Autodesk product. By the way, we made it.
[54]Decision on initial pricing of AutoCAD.

QB86

META is now running on the 8086 but has not been modified to give some features that the QBASIC compiler will need for compatibility with CB80. The corresponding compiler changes have not been coded.

Pass 2, the optimizer, now generates 8086 assembly language. Now that we have the definition of the relocatable format, QP2 must be modified to generate relocatable directly, as well as supporting the new CB80 compatible features.

The memory model is well defined, using the Large Model for 8086 addressing. Parameter passing is mostly defined, but not all decisions have been made.

Queue handling and buffer allocation are working. The floating point library is working after a pretty complete rewrite and will support 32 and 64 bit integers if we want them.

There is a large amount of work needed on the library, without enough people to work on it. This is also the most easily partitioned part of the product.

To get all the parts of the project moving, there was a general reshuffling of manpower. Dan Drake will do the compiler pass 1 and associated changes in Meta. Mike Riddle will rewrite pass 2 in C; optimization will be based on the current QP2, and the output will be relocatable. Greg Lutz will work on parts of the library.

We will have a working version for Victor at the beginning of October. The big question is whether Digital Research will get CB86 out before we get ours out.

Window

The in-core version of line database is running. There is disk code, which is not running yet. There are some problems with string linkage and inadequate documentation of CB80 linkages.

The consensus was that the Z80 version would be ready in a month. The manual update should be an afternoon's work. The 8086 version will be out when QB86 is.

Corporate

Stock options are being set up with corporate counsel. Most of the paperwork will be ready to execute at the next meeting.

The process of unanimous decisions was discussed briefly. The unanimous agreement procedure that we adopted was intended for the adoption of new participants, not for other corporate decisions that the management or the board can make. If the organization gets large, polls of everybody will get too awkward. As we don't expect to bring dozens of people in on the same basis as founders, this may never be a problem.

Appendix

Task Lists

We need to have lists of what everyone is working on, so that we know what's covered and what's being overlooked. Also, everyone should know where to go with new information or questions concerning any project.

There is now an MJK user name called TLIST which is to be used for sending the lists. By the Wednesday night before each meeting, please send TLIST a list of what you've accomplished and what you intend to have done by next month, on a fairly specific level. Please keep it compact, to avoid overflowing the number of lines that TLIST is allowed to have stored. On the Wednesday night before each meeting John Walker will pick up the messages and compile them into a document that everyone will get a copy of.

MJK Usage

On our last bill from MJK,[55] 35% of the computer time charge was for prime shift. We can save substantial amounts of our working capital by not using the system unnecessarily in the daytime. We are also running up charges for file storage. If you use MJK at all, please pick up and delete your messages.

Next Meeting

The next (September) meeting is at 1:00 on Saturday, August 28, at Jack Stuppin's house. There will be a demonstration of BitStik, a frighteningly good graphics package on the Apple.[56]

Agendas

We've been making some attempt to keep to an agenda at these meetings without suffocating in Robert's Rules of Order. The responsibility for the agenda has fallen on the corporate secretary, Dan Drake. It would be nice to know what people now think about how the meetings should be run. Should we drop agendas entirely? Should we be more authoritarian in holding to the agenda? Should the agenda be changed in form?

In any case, if you have something that you think ought to come up at the meeting, please send a message to Dan Drake so that it can be brought up at a reasonable point on the agenda. If it seems too small a matter to put on the agenda, you can bring it up under new business.

Here is the normal skeleton agenda:

[55] MJK is a company located in Silicon Valley that sells timesharing on HP minicomputers. Their major business is providing access to current and historical data for the commodities market. John Walker used this system in his commodity trading days, and also used it to implement a teleconferencing system for Marinchip dealers and, later, Autodesk people. Since the system was on Tymnet, overseas participants could access it economically.

[56] It was a demo of the BitStik in London in August that prompted John Walker into a furious burst of implementation to upgrade AutoCAD.

- Call to order.
- Introduction of any guests.
- Review of minutes.
- Financial report.
- Progress reports: Everyone gets 2 minutes to tell what's happening. More details will be covered in project reviews.
  - How much time are you putting in?
  - What have you been doing; what have you accomplished?
  - What will you be doing during the month?
  - What's standing in your way?
- Random new business and things that came up in reports.
- Project reviews.
  - Marketing
  - Autodesk
  - MicroCad
  - QB-86
  - Window
  - Corporate

AutoCAD-80 Development Log

During the summer and early fall of 1982, I was working furiously on AutoCAD-80, the CP/M-80 version of AutoCAD. At the same time, Dan Drake and Greg Lutz were working on AutoCAD-86 for the IBM (Greg) and the Victor 9000 (Dan). Because AutoCAD-80 started to work earlier, largely because it used intelligent display devices and didn't require the extensive low-level drivers that the IBM and Victor needed (and still need), AutoCAD-80 took the lead on feature implementation through the introduction of the package at COMDEX in November of 1982 and into early 1983.

What follows are excerpts from the extensive development log that chronicled AutoCAD's earliest formative stages. I've tried to select sections that show the first appearance of key facets of AutoCAD, foreshadow features implemented much later, and give a general flavour of the initial development of the package. You can see that from the very start a major theme in AutoCAD development was figuring out how to make it fit in memory. The version described below had to run in a machine with a total memory of 64K bytes, of which only 52K was free for user programs.

The log was begun in July of 1982. The practice of dating entries did not begin until late August.

MicroCAD Notes

When you specify a length (e.g., circle radius, size for text), GDATA/GEDIT should allow you to specify two points (if the first arg was a point, it would ask for a second. Then it would take DIST of the two points and use that for the length. If the second point were null, it would use the origin of the object being entered (center of circle, base of text, etc.)).[57] Installed in GEDIT 8/19/82. Could also do the same thing in GORIENT for angles, but it's a lot less frequently used and I don't feel like doing it at the moment.

Text and shape sizes should be specified in terms of the basic size of the drawing, not as a scale factor.[58]

To compress the redraw file: if start point of current coordinates is same as end point of last, just output (X2+10000, Y2).[59]

Group designation. When you're asked to "digitise entities", you should be able (somehow) to specify one or more boxes, as for the SOLID entity. Then every item which has a vector within that box (or is totally

[57] Origin of "two points" specification. This was the first user-interface improvement in AutoCAD.

[58] Text size was originally specified as a scale factor for the text definition size, which varied from font to font.

[59] This type of compression was not actually implemented until release 2.5 in 1986, though it was used much earlier for slide files.

within the box [which??]) should be selected. This will let you grab a group of stuff in a drawing and move it somewhere else, or delete a section. Current point selection is a special case of this.[60]

Speed up regen—for every entity we should be able to calculate an "enclosing box". This is a conservatively larger box which we know to contain all vectors generated by the entity. If the box is totally off the screen, we can skip regen of the entity entirely and avoid all the calls on CLIP. If the box maps into one pixel, just draw a dot and forget it. This should make large, complex, drawings viewed in small pieces much more efficient.[61]

We need an interactive shape editor.[62]

All versions of MicroCAD should be able to write an "entity interchange format" file. The utility which does this may not be actually in the main package, or may be called as an overlay. All versions of MicroCAD, regardless of internal file representation, will be able to interchange drawings this way.[63] Installed in MicroCAD-80.

Design change: Made DBLIST quit list and return to command mode if you type Control C. Fixed to also pause on Control S. Document this & pass on to C version.[64]

Can we use a better algorithm for FILL? Maybe we should have an entry in the display driver for FILL, as some displays have that capability. In fact, maybe we want to totally rework the interface between the display driver and MCAD: displays are getting smarter and smarter, even the Microangelo can draw circles, and the NEC can do arbitrary arcs as well as fill. Maybe the display should be passed an entity clipped and scaled to screen coordinates. The display driver could either do it itself, or buck it back to a vectoriser mechanism which would call the display again with individual lines as at present. Note that this would seriously mess up the REDRAW file interface and in particular VRTST, which would now have to test any kind of entity the device can draw.[65]

Dimensioning drawings—one should be able to specify a default scale for the drawing, e.g., feet, angstroms, kiloparsecs, and have all MCAD communications in that scale. MCAD should come with a units database, so if the user appends a unit name to a number input, it will perform the conversion to the drawing scale. You should be able to have a drawing rescaled to different units, or just ask for any output (DIST, AREA) in other units (for example, your design is metric but you need to know how many square feet of aluminum plate to buy).[66]

At some point we're going to have to put in some kinds of macro command facility. It might be wise to do this sooner rather than later, since we might avoid lots of custom coding and application specific stuff we'd otherwise get asked for. Obviously, the boundaries between shapes, INSERTs, and command macros are fuzzy. Here's an example of a request from an architect which I'd like to write a macro for. After finishing a floor plan, they want to put in those little dimension lines that look like:

---

[60] First mention of window object selection, and allusion to the distinction between window and crossing selection.

[61] First suggestion of maintaining entity extents and using them for quick rejection. Not implemented until release 2.5, June 1986.

[62] We still do, but not as much as when this was written, when blocks had not yet been invented.

[63] Original suggestion of **DXF** files.

[64] First cross-fertilisation of PL/I AutoCAD-80 and C AutoCAD-86.

[65] Hardware solid filling was implemented in release 1.2, and the software fill algorithm was eventually rewritten in release 2.0 (October 1984). Passing high-level drawing functions to the display driver is still a dream at this writing in 1987, for the reasons mentioned herein. *Note added in fourth edition:* Subsequent upgrades to ADI now allow a display driver to receive much higher level information about the geometry being generated and permit high-performance display list drivers to be created.

[66] Great idea. Never implemented. *Note added in fourth edition:* The implementation of "Physical Units" in Release 11 was the first step toward this noble goal.

and they’d very much like to do that just by designating the two end points of the dimension and where the legend was supposed to go. Try working out this example in your head, and you’ll see that some form of “entity variable” is required in the macro facility as well as control structures.[67]

Digitiser menu. We should supply preprinted digitiser menu overlays that the user could Xerox & write in the legends for his menu. When you install the program, you would tape the menu to the digitiser & digitise the two corner points. MCAD would then figure out where it was & therefore where the subdivisions were. We might also put the number of vertical & horizontal divisions in the configuration file, so you could make as crowded or open menu as you liked.[68] We might want to provide the option of a screen menu as well. This would be essential for use with a light pen, and might make the digitiser menu work better as well (wouldn’t have to take your eyes off the screen). Maybe this could be cleanly integrated with the split-screen work we’ll have to do to support one-screen systems.[69] If you have a large digitiser, you could have a small digitiser menu you could move to the area you were working on. Any designation within the menu would indicate the menu. To hit a drawing point there, just move the menu first. I installed a position sense and auto-scale in MCAD-80. When you specify a MENU file, it looks for the file **`MCADMENU.CFG`**. If this file is present, the digitiser menu lower left corner & upper right corner are taken from it, along with the default menu file. The menu is then automatically loaded. If the file does not exist, the user is asked to digitise the menu corners and to specify the menu file name.

Terminology change: SHIFT has been renamed PAN.

Terminology change: ARRAY is now REPEAT. REPEAT is now ENDREP.[70]

Terminology change: ORIGIN (for INSERT) has been renamed BASE. The BASE command now sets it, and it is called the “Insertion base” in the STATUS display.

Terminology change: SNAP is now RESOLUTION (which may be abbreviated to RES). It works just like it used to, but defaults ON at the start of a new drawing. The status display was changed to call it “Resolution”.[71]

Implemented a new option on the GRID command. You can now say GRID 5X for example, and set the grid to 5 times the resolution (formerly snap value). You can still specify a number without the X and set the grid to anything you like. Should we have grids with different X & Y increments?

Installed the new LIMITS facility. Drawing limits may be changed by the new LIMITS command. GPOINT will reject any point outside the defined drawing limits. Extents have been redefined to be the corners of a box enclosing the actual data in the drawing. They are recomputed on every REGEN (note special handling in REGEN for aborted REGENs). In STATUS they are displayed, flagged if the drawing runs outside the limits. ZOOM ALL was changed to display the drawing limits, or the extents, whichever is larger.[72]

[67] This suggestion foreshadowed the development of AutoLisp, released more than three years later.

[68] At this writing, the digitiser menu was fixed at 40 boxes at an absolute position on the tablet. This feature was finally implemented in release 2.0 (October, 1984).

[69] First suggestion of an on-screen menu.

[70] When REPEAT/ENDREP was eventually retired in release 2.5 (June 1986), the command that replaced it was called ARRAY. It’s all on the wheel; it all comes around.

[71] This was one of Stephanie Nydell’s suggestions. We changed it back to SNAP in release 1.3 (August 1983).

[72] First appearance of LIMITS. Fascist limits ruled until release 2.1 (May 1985), when the ability to turn limits checking off was provided.

The LIMITS facility has repercussions that run through MCAD. ZOOM ALL was rewritten again to hopefully do what we want. It gives you a display with the lower left corner aligned at the X and Y coordinates of the left lower drawing limit or extent, whichever is less. The extents were changed again to be reset based on generation only for a ZOOM ALL which runs to a normal completion. That means that if you delete something which reduces the used space extents, they won't be updated until you do another ZOOM ALL. Any other approach I can think of has disastrous implications on any attempt we might make to optimise generation of entities. See the comments in DSCMDS and CSCALE which explain this change.

Removed the CENTER command. It has been superseded by the ZOOM C option.

Installed the ZOOM L option. This allows setting the screen window by the lower left corner point and the side width.

Made the LAYER command accept COLOUR as well as COLOR for our friends across the Atlantic.[73] Note that since we've replaced CENTER with ZOOM C, center/centre is no longer a problem. I may just change LAYER to look at the first 2 characters of the command, but I haven't yet.

Added the OOPS command. Now whenever ERASE is erasing entities, it makes a list called OOPSLIST which records the entity database file location of each entity erased. OOPSLIST is cleared at the start of an ERASE command, so following an ERASE, OOPSLIST represents the entities deleted by the most recent ERASE. The OOPS command scans OOPSLIST and goes through the entity database "un-erasing" all the entities on the list (this is easy because we erase simply by negating the TCODE of the entity, to bring back, just negate again).

The problem of REPEAT/ENDREP accumulation in files could be eliminated by detecting an REPEAT immediately followed by a ENDREP and deleting both on the fly. Then if you delete all the items within the pair, the REPEAT block would go away as well. Not hard to do on a REGEN.

Might we want to install a MODIFY command? It would work just like a LIST, but would give you the chance to change entity properties. Or is the ability to edit the drawing interchange file enough (I vote no). Any suggestions on how to specify a MODIFY of an ENDREP?

I went at the digitiser interface with a chainsaw on 8/26/82. The changes are major in concept and scope, and will be described below:

First, to allow greater resolution for larger digitisers, the scale of coordinates returned by DGDRV was changed from 0–1023 to 0–20479. This twentyfold multiplication of scale allows us to lose no resolution on a digitiser which resolves 200 points per inch and has a longest dimension of 100 inches. So much for large digitisers.

Second, I changed the way DIG smooths samples from the digitiser. Previously, it waited until it had two samples and averaged them. I changed this to use an exponentially smoothed moving average with smoothing constant .10 (it can be changed in DIG [parameter SMOOTH]). This technique (used to smooth radar samples when computing trajectories) enormously reduces the jitter caused by single random samples, and imparts a "buttery smooth" motion to the cursor on the screen regardless of how jerky the digitiser sampling rate or motion is. It also avoids the overflow which would result from adding two FIXED(15) numbers in the 20000 range.

Third, I installed a totally new mode where the digitiser is used as a true digitiser rather than a screen pointing device. When you enter MCAD-80, the digitiser works just as it does in INTERACT. If you enter the TABLET ON command, MCAD-80 asks you to digitise two points on the drawing on the digitiser and enter

[73] First zephyr foreshadowing the coming hurricane of language translation issues.

their drawing coordinates. From the digitiser coordinates and the coordinates entered, MCAD-80 computes the scaling, translation, and rotation of the coordinate system of the digitiser with respect to that of the drawing, and saves these parameters. It turns on TABLET mode.[74] In TABLET mode, whenever GDATA processes a digitised point, it will transform the digitiser coordinates into the drawing coordinates. This allows dimensioned material to be entered and the actual drawing coordinates to be stored in the entity database.

This is completely general: the full resolution of the digitiser is used in the calculation, and the drawing may be placed on the digitiser in any orientation; may have any desired coordinates (we assume the coordinates are rectangular), and any two points may be digitised to establish the transformation to digitiser space. Thus a small drawing may be placed anywhere on the digitiser, and a large drawing may be digitised in pieces.

When in TABLET mode (which is indicated on the status display), points from the drawing are mapped into drawing space without regard to what the screen displays. Points may be entered which are off the screen, and the screen resolution is of no import. In TABLET mode, the cursor on the screen will track the digitiser cursor as before, but the relationship between where it points on the screen and the drawing points is broken (see below for why this done). The menu may continue to be used while in TABLET mode. Entities entered will be displayed windowed to whatever the screen window is.

Those commands which use entity designation rather than coordinate specification (such as ERASE, MOVE, and LIST), may be used while in TABLET mode, as long as the entity being specified is on the screen. In this case the cursor is used to point to the item on the screen. (That's why it displays in screen coordinates even though TABLET mode is on). The normal mode one would use when entering data in TABLET mode is to set up the TABLET, then do a ZOOM W to put the area being entered on the screen (just point to the drawing areas, TABLET mode will worry about internal coordinates). I think this is reasonable, but it isn't hard to make the cursor track the drawing coordinates rather than the screen if that seems better. I just didn't want to give up the ease of entity designation while in TABLET mode for the common case of deleting a bad entity just entered. My feeling is that the cursor position doesn't matter in TABLET mode because the user is looking at the digitiser, not the cursor on the screen.

The TABLET command has other subcommands. TABLET OFF turns off TABLET mode and restores normal digitiser operation. A subsequent TABLET ON command will turn TABLET mode back on with the same coordinate transformations as before. TABLET CAL (for CALIBRATE), forces a recalibration of the tablet, in case the drawing is moved. TABLET CAL turns on TABLET mode if completed successfully.

Changed the extension and nomenclature for interchange files. Previously they were "Entity Interchange Files", **.EIF**. Now they are "Drawing Interchange Files", **.DIF**. There's a hundred people who know what a drawing is for every one who knows what an "entity" is.

On 8/27/82 I installed the following optimisation for text generation. If the text is less than 4 dots and more than 2 dots high, I just draw a dot for each character centered vertically in the text height, with each character using .6 of the text box width (who knows?). If the text is less than 2 dots high, I just draw a line where the text would go with length equal to $0.6 \times txsize \times length_of_text_string$. If the text start point is above the top of the screen or to the right of the screen, all generation is skipped. If the start Y coordinate plus the size of the text is below the bottom of the screen, we skip generation. Also if the start X coordinate plus the string length times the height is less than the left of the screen. The rules for skipping generation entirely are pretty conservative and shouldn't lose text unless you're doing something funny. See EREGEN for full details on the

[74] This was, to my knowledge, the first piece of totally general Euclidean geometry installed in AutoCAD. I woke up after four hours of restless sleep with the constructions for the TABLET command fully formed in my mind and a compulsion to implement it. Geometry does that to you.

horrors arbitrary text rotation introduce in implementing these tests.[75]

Here's an idea. We've all come up with the idea to optimise regen by calculating an "enclosing box" for every entity. The problem is how to calculate the bounds of the box for shapes and text (arcs don't come cheap either). 'Spose, however, that we add 4 cells to each entity record. When we EREGEN the entity the first time, we have CLIP calculate an entity extents & save them in the entity item. Once calculated, we can instantly see whether an entity is on the screen or not. The disadvantage of this is that it makes the entity file bigger by 4 reals per entity; this is significant, and I don't know enough to say whether the speed is worth the space. It wouldn't be horrible to "have it both ways" I guess. Note that this scheme also gives you a very cheap test for whether the entity contains enough detail to bother generating.[76]

As you know, I have great plans for INSERTs. One of the dilemmas with INSERT is that it can be used both for combining drawings done in pieces and as a user-defined part facility. When combining drawings, you want the layer information in the INSERT to be preserved (which is what INTERACT currently does), but when you use an INSERT as a part from a library, you want the INSERTed entity to go onto the current drawing layer. Rather than muck things up with modes and commands, I have installed the following logic: entity layers are copied on an INSERT *unless* the layer number from the insert file is 127. Layer 127 will be replaced by the current layer of the drawing. Thus, you can have it either way you like, and you define how the INSERT works when you create it (which is probably the time that makes most sense).

Further work on INSERT. When you load an INSERT, I now allow you to specify an arbitrary rotation, X size, and Y size.[77] The X size and Y size are divided by difference in the high limit and low limit, respectively of the X and Y dimensions of the drawing being inserted, resulting in X and Y scale factors to be applied to the drawing inserted. The rotation is used to rotate the inserted entities around the insertion base of the drawing being inserted. The resulting transformed entities are then inserted in the working drawing as before, with the insertion base of the inserted drawing aligned above the insertion point designated. If the X and Y scale do not result in a rectangular coordinate system for the inserted drawing, the following restrictions apply:

- CIRCLEs will have the centre moved correctly, but will not be displayed as ellipses. The X scale will be used to change the radius.
- ARCs are totally messed up. I should turn the internal representation back into 3 points and generate a new arc from the points. Instead, I just move the centre, scale the radius by the X scale, and rotate the start and end angles. Doing it right is intended to be in the final release. It isn't hard, just complicated.
- The height of TEXT is always scaled by the Y scale.
- SHAPE size is scaled by the X scale.

Note that everything works correctly as long as the X and Y scale are equal. I envision the installation of a new entity called a SCALE FACTOR which handles aspect ratio changes and scaling at the CLIP level. Then an INSERT can just generate this item if required. That's why I didn't go to great lengths here. As long as the drawing contains only lines, points, and traces, it may be changed in aspect ratio without restriction. Anything

[75] This the first entity generation optimisation ever installed in AutoCAD.

[76] When we finally installed the index file (cheat file) in release 2.5, this is almost precisely how we did it.

[77] As you'll see below, this implementation was discarded in less than two weeks in favour of the far more general INSERT mechanism still used today. I'm including this description of the initial implementation of INSERT to give the flavour of the many pieces of code implemented and then discarded in favour of more general solutions.

may be rotated without restrictions. Note that you can rotate your whole drawing by starting a new drawing and INSERTing the old one with a rotation specified and unity scale factor.

I also threw some effort into civilising the LAYER command. Now when you say LAYER, it prompts you with "Layer (Layer no./ON/OFF/COLOR): ". If you say ON or OFF, it prompts you for the list of layers (see **LAYRCM.PLI** for exact messages). The LAYER command previously did a REDRAW on normal exit, because it may have changed the visibility or colour of the lines on the screen. If LAYER exited in error (for example because a bad number were entered after an ON, this REDRAW would be skipped). I changed it so that whenever you leave LAYER, you get a redraw regardless of what you've done. Thus if you say: LAYER ON 1,2 OFF 19 FOOEY, the screen gets updated to reflect the changes you've made before the error.[78]

Installation of the menu driver and associated sicknesses: now **MCAD.COM** is the program you call to enter MicroCAD. You can optionally specify a drawing name on the call. If you do, that will be the default drawing name for menu selections. LPROG was changed to accept an extension, and all the other actual main programs (**MCADE**, **MCADP**, etc.) were given extensions of "**.OVL**", which will prevent some gonzo calling them directly from the console. **MCADE** always works on a **.$$$** work file. This file is prepared for it by **MCAD**, and is converted back into a **.DWG** file or discarded after **MCADE** returns with its completion code in the command tail. I added a PLOT command in **MCADE**. This chains to the plotter driver, which produces a plot of the current state of the **.$$$** drawing file, and chains back to **MCADE** right where you left off. This is very handy for interim views of the drawing.[79]

On 9/6/82 GEDIT was enhanced to allow you to specify two points when an angle (GORIENT) is required. It takes the angle between the first and second point and uses that for the orientation. That way, you can "show it" which way you want to text etc. to run.

As suggested by Richard Handyside, I made entering just an "**@**" when a point is expected return the same point as last entered. This works out as a logical default, since **@***x,y* is the relative point specification.

Totally re-did the INSERT mechanism. These changes supersede the INSERT enhancements mentioned above, offering all the capabilities with none of the restrictions (the previous INSERT was done as a test vehicle for parts of this implementation). Three new entity types have been added. Type 12 is "Block definition start" and has a name, and X and Y base as attributes. Type 13 is "Block end" and has no attributes. Type 14 is "Block reference" and has attributes of block name, X and Y position, X and Y scale, and rotation.[80]

When you do an INSERT, MCAD asks you for a block name. If this block is not already used in the drawing, it loads the block from a file with the same name (a different name may be used by saying *blockname***=***filename*). Drive specifiers will be stripped from the file name if it is used as the default block name. A block definition start entity will be placed in the file, with X and Y base drawn from the master record of the drawing being INSERTed (old terminology, origin). Then all the nondeleted entities will be copied into the current drawing, followed by a block end entity. Unlike the old insert, the coordinates of the drawing being loaded are not translated as the entities are loaded. They remain in their host coordinates forever. All transformations are done when the block is elaborated as a result of a block reference entity.

Regardless of whether the block was previously defined or just loaded, a reference to it will then be generated, with the user being prompted for insertion position, X and Y scale, and rotation. As X and Y scale default to 1, and rotation 0, the effect of the command defaults to the operation of the old INSERT.

[78] You might call this the first of the many rewrites of the LAYER command.

[79] This marked the introduction of the "AutoCAD Main Menu" which has proved so durable. Up to this time, the drawing editor and plotting module were separate programs called from the operating system command line.

[80] This is essentially the block mechanism used today.

After entering the entities, PREGEN is called as before to process them. EREGEN contains almost all the support for the new entities. A block definition not previously seen causes the block to be scanned to the matching end (oh yes, blocks can be nested without limit other than memory space). The block is then defined on the in-memory list, with the start and end entity locations saved. If a block definition is seen for a block already known, the block is immediately skipped by the expedient of plugging the already stored end entity location into the entity scan address. This means that having lots of long block definitions in your drawing doesn't slow you up on REGENs.

When EREGEN sees a block reference entity, it makes a logical subroutine call in the entity file, pushing the return entity address and continuing with the first entity of the block referenced. Based on the attributes of the block reference item, a transformation descriptor is constructed and stacked so that CXFORM will correctly map the internal coordinates of the block being elaborated into the location, scale, and orientation desired by the user. Since both the return and coordinate transformation items are stacked, block calls may be nested without bound. EREGEN will always treat the regen of a block reference item as primitive. Hence one call on EREGEN may actually draw hundreds of entities. NOTE that as a result, the entity in the entity record when you come back from EREGEN may not be the same one as when you called it. Hence, people like MOVE, and OOPS had better do their MODDR before, not after, the call on EREGEN.

When CLIP is drawing vectors and writing them in the refresh file, it tests whether any block elaboration is underway. If so, all the vectors in the refresh file are tagged with the entity location of the OUTERMOST block being elaborated at the time. Why outermost? Because if you INSERT a part with complex internal structure, and you happen to point to an internal part, you still want the whole thing to move (be deleted, etc.). This very simple trick makes INSERTed entities primitive to all MCAD operations.

But why, you ask, copy the INSERTed entities into the drawing? Why not just save the block name as a file reference? I decided to copy them for reasons of efficiency and maintainability. CP/M file opening is extremely slow—we cannot tolerate a file open for every block reference, so we'd have to open the files and leave them open—not attractive from the standpoint of buffer space! Also, I feel that it's a valuable feature that a drawing be self-contained. If INSERTs stored references to other files, the user couldn't just copy the drawing and be confident he had everything. It would also make user programs which process **`.DIF`** files much more complicated.[81] If the user wants to change the definition of the part, we can easily provide him the option when the temp copy of the drawing is made (in the menu driver) to replace one or more INSERT blocks in the drawing with new versions in files.[82] Thus, after long thought, it seems to me that the way I put it in is the correct way from a design standpoint, not considering implementation at all. It certainly would have been easier to make each INSERT scan a file! Also, the way I did it allows us to let the user define his own blocks in the drawing independent of INSERT files. This might be handy in certain drawing environments.

Hence, with these changes to INSERT, the art of coding SHAPEs will become much less necessary for most users. This gives us every capability in accessing stored drawings that Robocom has, and more (as their system cannot store parts which contain other parts).

Here's a point of contention: should a GRID be drawn by setting the grid points on, or by inverting them? If you had a drawing with lots of SOLIDs, the grid might be hard to see. On the other hand, grid points would seem to break lines into separate segments. I've left it as always turning the points on, but I'd like comments on which it should do.

Freeze for stabilisation declared on 9/14/82 04:30. No additional features to be added until release of level 1.0.

[81] This is still the essence of the justification for copying block definitions into the drawing that use them.

[82] Block redefinition was eventually implemented in release 1.4 (October 1983).

Only discrepancy resolution form[83] changes will be made.

DRF#1 9/15/82 1:17 Arcs and circles don't always regen if part of a block which has been scaled. Corrected by applying all scale factors to a point centerX+radius,centerY+radius, taking DIST of that, and using that for the test for on-screen. Some extremely bizarre cases may still fail, but no realistic ones. For example, X scale=20, Y scale=.1, angle=90. Is it worth calling CXFORM 4 times to fix this?[84]

DRF#2 9/18/82 2:09 No way to configure a system without digitiser without accessing I/O ports. Output ready mask for digitiser (otherwise unused) was defined as "digitiser present flag". If nonzero, digitiser port will be polled. If zero, digitiser read routine always returns "no sample" and no I/O port accesses will be made.

DRF#3 9/18/82 2:12 Bad test for drawing limits in GEDIT. Fixed.

DRF#4 9/18/82 2:04 Can't set high drawing limits to large (> 600) value. If an overflow happened in CVDTS in attempt to BLIP point, coordinates weren't returned. Changed in GEDIT to ignore overflows in attempt to BLIP (since it's off screen, BLIP wouldn't do anything anyway).

DRF#5 9/18/82 13:28 Setting resolution to zero in MICROCAD doesn't turn off resolution snap mode as documented. Changed to do so. Also, if resolution is set to zero, default GRID value to X limit/10 to avoid confusion if GRID ON is done.

DRF#6 9/18/82 13:26 Mike Riddle points out that "**.DIF**" is used for VisiCalc interchange files. Changed drawing interchange file extension to "**.DXF**" to keep some gonzo from trying to load one into VisiCalc.[85]

DRF#7 9/20/82 16:16 Shape compiler wasn't closing output file before chaining back to the main menu. This caused the last block not be written out. Changed to close output file in every case. (Reported by Jamal Munshi).

AutoCAD-80 Release 1.1 Development Log

On 10/10/82, I changed ERASE and MOVE to remove the old item by drawing over it with dark vectors. This eliminates the pesky and time-consuming REDRAW which used to follow every ERASE or MOVE, and makes things run immensely faster.[86] The routine UNDRAW in REDRAW accomplishes the magic. Note that if you have overlapped entities, the process of undrawing an entity may turn off bits which should be set by other entities. If this happens, and you really need to see a cleaned-up screen, you can just type REDRAW and everything will be correct. As with the use of FLOOD, my feeling is that this is the correct choice as the full REDRAW was just intolerable for complex drawings. This code has been integrated with the flood code for TRACE and SOLID entities, so that completely filled TRACEs and SOLIDs will be flooded with zeroes to turn them off. The Microangelo provides no way to turn off a pattern-flooded area, so such an area will remain on the screen until you do a REDRAW (its boundary lines will be turned off, making it clear the entity has been deleted).

I went through the whole thing and changed it to AutoCAD. This took about 6 hours because all the MCAD

[83] Like a proper software company, we had to come up with a suitably pompous name for a bug report. Only years later did we actually start calling it a "bug form".

[84] The very first AutoCAD bug ever reported, and the first fixed.

[85] Thus, **.DXF** was chosen as the extension for drawing exchange files.

[86] That's right, up to this point every command that erased something from the drawing completely redrew the screen rather than erasing the object with dark vectors. This was visual fidelity carried to the lunatic fringe.

segment names buried in more than half of the modules had to be changed, and the modules recompiled. I sure hope this is the last time! You now start it up with ACAD, and all the file names which previously contained MCAD now use ACAD instead.[87]

DRF#17 10/11/82 19:53 Some people thought PAN worked backward, so it was changed to work backwards. (Reported by Keith Marcelius).[88]

I revised the interaction of INSERTs and layers. Previously, INSERTed entities retained their original layer, except entities on layer 127 were statically moved to the current layer at original transcription time. This was not very useful. I changed it so that an INSERT's layers are preserved, except that any entity in a block with layer of 127 will be drawn on the layer of the outermost block active at the time the entity is drawn. This lets you define an insert and put it (or parts of it) on any desired layer at the time it is used.

All the new features contributed to memory growth which blew off our goal of running in a 52K user space. To free up more memory, EACQ, which was the longest overlay, was split up into two separate overlays (ACAD9=EACQ1, ACAD10=EACQ2). SHAPE, TEXT, TRACE, and SOLID acquisition were moved to EACQ2, and COMMAND was changed to call the correct overlay based on which entity type is being entered.[89]

On 10/29/82 I installed light pen support code. There are two new configuration variables for the Microangelo associated with the light pen. The first is called LPUSED, and should be set to **OFF** if the light pen is used and zero otherwise. The second is called LPDELAY, which controls the light pen selection logic. When the light pen is enabled, the screen is run in reverse video so that the tracking cross may be seen better. To designate a point with the light pen, you "pick up" the tracking cross, move it to the desired point, and let go of the tracking cross (by removing pressure from the light pen or taking your finger off the end region). After the delay specified by LPDELAY, the point will be selected. Be careful not to move the light pen too fast across dark regions of the screen, or you'll lose the tracking cross and incorrectly designate a point within that region. Both a light pen and digitiser may be used on the same system, but only one at time. If both are connected, the digitiser has priority; to use the light pen, just remove the digitiser cursor from the tablet (or otherwise make it stop sending samples).

On 10/30/82 I installed code by Greg Lutz to correct a bug he found in the interaction between INSERTs and REPEAT/ENDREP loops. If an INSERT was invoked inside a REPEAT loop, the coordinates of the block invocation entity were transformed by ETRANS to the loop instance coordinates. When the contents of the block were read, ETRANS transformed them again, resulting in (in the simplest case) double translation of the entities. The new code makes block invocation push AXTRANS and AYTRANS and set them to zero before a block is elaborated. At the end of the block, they are restored from the block execution (BX_) item. This should make all combinations of blocks and repeats work correctly.

Corrected a bug in TRACE entry reported by Richard Handyside. While entering a continued TRACE, if a point outside the drawing limits is inadvertently entered, a bizarre last leg of the trace was drawn. The TRACE entry code was assuming that when GPOINT returns an NVALID result, the X and Y coordinates were unchanged. In the case of an out of limits point, this is untrue (so that ID can be used on out of limits points). As it turned out, TRACE had already copied the last point to separate variables, so all that was needed was to use them. Now entering an out of limits point will terminate a trace just like hitting the space bar (that is, putting a right angle end on the last segment of the trace).

[87] The product becomes AutoCAD, and **ACAD** is used to call it for the first time.

[88] And backwards it still was. After this change was panned by the critics, we changed it back yet again in release 2.0 (October 1984).

[89] First restructuring of the program to make it fit in memory.

If the drawing editor crashed while editing a drawing, all changes made in the editing session would be lost beyond hope of recovery. This is because CP/M only updates the file directory item for a file on the disc when the file is closed, and even though we were faithfully writing out the new entities and refreshing the drawing header on every REGEN, all this new information would be lost if the normal closeout code were not executed. I changed REGEN to close and re-open the drawing file at the end of every REGEN. This means that if ACAD crashes, all you have to do to get back to the point of the last REGEN (ZOOM, PAN, ENDREP, or anything else that causes a REGEN) is rename the **.DWG** file to **.BAK** and the **.$$$** file to .DWG. The **$$$** file thus contains a valid drawing as of the last REGEN. If you're doing a long editing session and want to make sure you're protected, just do an explicit REGEN every now and then. In normal use of the package, the user will probably be doing enough REGEN inducing operations that loss in a crash will be minimised.[90]

On 11/1/82 I made the ZOOM and PAN commands in DSCMDS an overlay (**ACAD11**). This freed up about 1600 bytes of memory to waste on other features.

AutoCAD 1.2 Development Log

On 11/17/82 I installed an on-screen menu for the light pen. This works the following way: if configured, the right border of the screen is dedicated to a menu, with one line for each of the 40 menu items. When you point at a menu item, it flashes. You may move up and down the menu area and the flashing will follow you to confirm the location. When you get to the desired menu item and release the light pen, the menu item will go to reverse video as it is executed. It will be restored to normal when the light pen is next pointed into the menu area. When the light pen menu is used, the aspect ratio of the screen is changed and XDAR and YDAR are adjusted to accommodate the space removed from the right of the screen.[91]

On 11/18/82, as requested by Lars Moureau and Keith Marcelius, I added a feature to INSERT which allows old-style pure transcription of the file being loaded, rather than copying it into a block. You select this mode by preceding the file name with an asterisk.[92] You may insert a file previously loaded as a block without any duplication or error. The block is ETRANS relocated to the desired insert point based on its insertion base, but scaling and rotation are not allowed and the prompts for them are not issued.

To allow clearer and more concise labeling of MENU items displayed on the on-screen menu, I added a "menu label" field to the menu file. If the first character of a line in the **.MNU** file is "**[**", the text between the "**[**" and the next "**]**" will be displayed on the screen menu. The actual text sent to the input processor when that item is selected will be the text that follows the "**]**".

Often one wants to specify high-level commands using the menu facility but is thwarted by the inability to allow user-specified parameters in the midst of a canned input stream. No more. I installed a "menu macro" facility which allows the interpretation of the menu text to be suspended, a user input to be accepted (from the keyboard, digitiser, light pen, etc.), and then the menu text interpretation to be resumed. As the menu text is scanned, if a "****" character is hit, the next user input is accepted via GDATA(0) (that is, terminated by space, semicolon, or return), and logically inserted in the menu string at that point. After the user entry is processed,

[90] That's right, as of this point AutoCAD-80 had the ability to recover from a crash and restore the drawing to the last REGEN. Unfortunately, the in-memory paging of AutoCAD-86 precluded an efficient implementation of this capability and, to this day, AutoCAD-86 lacks an automatic crash recovery facility.

[91] This marked the initial implementation of the screen menu at the right side of the screen. It was initially intended as a convenience for light pen users, but rapidly became the primary means by which users interacted with AutoCAD.

[92] Thus, **INSERT ***.

the scan of the menu text resumes after the "****".[93]

For example, suppose we've defined an INSERT file called **NANDGATE** and we want to be able to insert it at 1×1 scale with no rotation simply by pointing to the location. We would define a menu item as follows:

```
[NAND]insert nandgate \1 1 0
```

This would do the INSERT command and supply the file name. Then a user input would be requested to supply the insertion location. The scale and rotation queries would be supplied by the menu text automatically. If that's not esoteric enough, define an insert called **BOX** which is a square with side of 1. You can then define a menu item as follows:

```
[BOX]insert box \@ \@ \0
```
Obsolete—see below.

To use this menu item, you select it, and then digitise the location you want the box to go. Then the "**@**" forces a "2 points" entry form for the X scale. You supply the second point from the digitiser which sets the X scale to make the box as wide as the distance from the box origin to the second digitise. The second "**@**" forces "2 points" for the Y scale also, and you enter the second point specifying the height. Since the box has side of 1, the scaling makes it as big as the lengths you've specified. The angle is forced to zero. Thus you point to BOX, point to where you want it to go, point to where you want the right edge to go, and point to where you want the top to go, and there's your box. And all without adding any new entities to ACAD! Yes, I know it would be nicer to just point to the upper right of the box, and I'm working on that.

Note that in defining these menu items, the placement of spaces to force execution of commands is critical. I'm downright embarrassed at how little code it took to put this in. It's all in GDATA and it's not much.

On 11/19/82 I added the ability to specify both the X and Y scale factors of an INSERT by simply entering one point. If you respond to the "X scale factor" query for an INSERT with "CORNER" ("C" is enough), then it prompts you for the "upper right corner". You digitise a point, and the $|\triangle X|$ becomes the X scale factor and the $|\triangle Y|$ becomes the Y scale factor. Thus if you've defined the INSERT to be 1 by 1 you can insert it and scale it to the screen just by digitising two points.

Thus, with the changes made today, the BOX macro defined above becomes:

```
[Box]insert box \center \0
```

To use it, you just point to the BOX menu item, the left lower corner of the box, and the upper right corner point. A box is then inserted to the scale you selected. To make the rotation variable, replace the last "**0**" with a "****", and you'll then be able to point to the angle you want the box to go on.

On 11/20/82 I finally got around to installing a buffering routine in LOAD. This cut the time needed to load the text shapes (**TXT.SHP**) by a factor of four. This reduces the annoying delay during the initial drawing of a picture while AutoCAD loads the text definition. I also cleaned up the carriage return/line feed logic of error messages in LOAD, which had a bad case of conceptual acne.

I made some minor changes in the format of the on-screen menu which allow the menu to be overwritten with a new menu without confusing the light pen logic. This lets you have items in your main menu which select subsidiary menus. (Previously if you did this, it confused the logic which inverted the last-selected item, and you wound up with two items inverted. These fixes correct this bug.)

[93] Menu macros are born. Until the release of AutoLisp in with release 2.1 in May of 1985, this was the primary means of extending AutoCAD. The initial implementation of AutoCAD AEC (AE/CADD) was done entirely with menu macros.

As suggested by Mike Ford, I changed the "Digitise entities" message to "Select objects".

I installed the long-awaited "window designation" facility to EID. The EID prompt message was further changed to "Select objects or window:", and EID was changed to recognise the letter "W" as a response before the first point is entered. Upon receiving the "W" it prompts for a left lower corner point and a right upper corner point (actually, you can enter them in any order). It then scans the refresh file and selects all entities which have at least one visible vector within the window, and from those only the ones which have no visible vectors outside the window. Read that over again and think about it for a while; it's not easy to comprehend, but it's easy to use and meets the most logical user assumption, I think.[94]

If you're using a light pen, often you want to move the tracking cross somewhere else on the screen to unclutter an area you're examining. Since the tracking cross only moves when the light pen is selected, and deselecting it designates a point, this generated an error message when the point was input to the command prompt. I changed COMMAND to just ignore points input at the command prompt, so you can move the tracking cross at will. Yes, I know, but this is the kind of little thing that users appreciate.

I added a "Last entity" selection option in EID. If you reply to the EID "Select objects" prompt with "L", EID will choose the most recently entity (whether visible or not). This is very, very handy especially for deleting the last thing you entered by mistake. Note that with ERASE you can step back through the file entity by entity by entering multiple ERASE L commands.

I installed Dan Drake's analytically correct code for calculating the number of segments to draw in a circle. This makes small circles smoother and big circles faster. It also optimises circles too small to see into a dot, speeding up REGEN of circles when you've zoomed way out. *Note to myself*: Dan's code uses ACOS, which pulled in 500 bytes of library. I must change it and define ACOS in terms of ATAN which ANG already pulls in. If you don't see a note below which says I did it, please remind me.

On 11/21/82 I looked up ACOS and redefined my own in EREGEN in terms of ATAN and SQRT. It gets the same answer as the one in the library and gets rid of 500 bytes (n.b., I wonder how much more can be saved with tricks like this—look at the library some time).

I installed a new BLOCK command. This command lets you create a block "on the fly" from parts of an existing drawing. This makes for more spontaneity while drawing, as you don't have to define all your parts ahead of time so you can insert them from files. When you enter the BLOCK command, you will first be asked "Insertion base:". Supply the point which is to become the insertion base of the new part (point to it on the screen). This has exactly the same effect as the BASE command when making an INSERT file. Next the standard "Select objects" prompt will appear, and you may use any of the entity designation options to choose the entities which are to make up the new part. To confirm that the entities you've selected are the right ones, they will disappear from the screen. Next the prompt "Block name:" will appear. Simply enter the name you want the new block to have (as no file name is relevant here, no equal sign should appear in the prompt). AutoCAD will then construct a block with that name containing the designated entities with the specified insertion base. You may then immediately use that block just like it was INSERTed from a file. The entities which were placed into the block are deleted from the drawing. If you don't want them to be deleted, just say "OOPS" after the BLOCK command finishes, and they will be restored to the drawing.

I installed a CHANGE command. This command is not the same as Mike Riddle's, and I don't know whether it is what we want. It is what I wanted to clean up drawings, and I'm convinced that something like it is a valuable addition to AutoCAD. I'm equally sure that what we really want isn't exactly this command, so

[94] Window selection is implemented. Its complement, Crossing selection, was not implemented until release 2.5 (June 1986).

maybe we don't want to document it until we come up with a final design. It is, however, awfully doggone handy. But on to the facts. When you say CHANGE, the first thing that appears is the standard "Select objects" prompt. You designate the objects that you want to change. Only lines and circles can be changed. Any other objects pointed to will not be changed (but they will disappear from the screen, only to reappear on the next regen—this is what we call "el buggo"). Assuming you've selected one or more entities, you will next be asked for an "Intersection point". If the object is a line, its endpoint closest to the intersection point will be changed to meet the intersection point. If ORTHO mode is on, the line's new coordinate in the direction of its longest run will be forced to the value of the corresponding coordinate of the other end. Thus, in ORTHO mode, only orthogonal lines will result from a CHANGE (this sounds clunky, but just wait until you use it to make a horizontal line meet a circle while zoomed way in, then you'll stand up and applaud). If the entity being changed is a circle, its radius will be adjusted so that the circumference of the circle passes through the intersection point. We all know that we need a much more general EDIT command which can talk about any property of an entity, plus modify entities based on properties of other entities (e.g., "Run this line in this direction until it hits that arc[95]" or "Make this circle pass through the endpoint of that line[96]", and so on), but until we get one, this thing makes cleaning up perspective drawings and hybrid digitised stuff like the shuttle drawing about ten times faster than before.[97]

On 11/23/82 I fixed a bug in EREGEN. If we were generating a circle or arc which was very large with respect to the screen, the code which calculated the number of segments to draw could divide by zero. Since in this case we want to restrict the number of segments to the maximum anyway, I just changed it to catch the ON condition and set the number of segments to the maximum, which is shorter in code and less complex than testing for zero everywhere in this fairly involved code.

The code which decides how many segments to use to approximate an arc or circle was basing the number of segments drawn on the relative sizes of circle and screen, but did not reduce the number of segments when drawing an arc. Hence, a 5° arc would be drawn with as many segments as a 360° circle. This made for very beautiful arcs, but very slow drawing! I changed it to first calculate the number of segments it would be drawing were the circle complete, then reduce the count by the expression $\max(2, NS \times TINCR/2\pi)$, which guarantees that no arc will be compressed to a straight line, but proportionally decreases the sector count for arcs.

I made another speedup change in EREGEN: previously I drew text as dots if it was too small to read and just as a line if it was smaller than a dot. I changed this to always draw text too small to read as a line the approximate length of the generated text. This makes display of small text about 5 times faster, as only one entity is generated per text entity rather than one per character. I find the result just as aesthetic. Comments are welcome.

[95] Finally provided in the EXTEND command in release 2.5 (June 1986).

[96] Provided by object snap in release 2.0 (October 1984).

[97] The CHANGE command was initially implemented to clean up drawings before Comdex 1982. I always viewed it as a short-term stopgap until a better editing command was developed. It's still with us.

COMDEX 1982

"Strange memories on this nervous night in Las Vegas. It seems like a lifetime, or at least a Main Era—the kind of peak that never comes again. Maybe it *meant something*. Maybe not in the long run... but no explanation, no mix of words or music or memories can touch that sense of knowing that you were there and alive in that corner of time and the world. Whatever it meant....

"There was madness in any direction, at any hour. You could strike sparks anywhere. There was a fantastic universal sense that whatever we were doing was *right*, that we were winning.

"That, I think, was the handle—that sense of inevitable victory over the forces of Old and Evil. Not in any mean or military sense; we didn't need that. Our energy would simply *prevail*. There was no sense in fighting—on our side or theirs. We had all the momentum; we were riding the crest of a high and beautiful wave."

Hunter S. Thompson, *Fear and Loathing in Las Vegas*

AutoCAD-80 Release 1.3 Development Notes

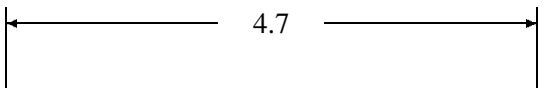
On 12/16/82 I installed a centered text option in EACQ2. When the "Insertion point" prompt is issued, you may now enter "C". This will cause a "Center point" prompt to appear—a point is supplied as for the insertion point, but the text will be centered both vertically and horizontally around this point. This is accomplished by making a dummy run through SHDRAW for the text with a new flag set which causes SHDRAW simply to update a max and min X and Y area used by the text. This area is then used to adjust the insertion point so that the text will be centered around it. Implementation note: we assume in this code that the text font has already been loaded before the first text entity is entered. Because earlier logic only loaded the text font the first time EREGEN encountered a text entity, this assumption was not always true. As a result, COMMAND was fixed to check whether the text font was loaded when a TEXT entity command is encountered. If it is not (TXGRID=0), then LOAD is called to load **`TXT.SHP`** before EACQ2 is called. This guarantees that EACQ will be able to calculate the size of the text if required for centering. The centered text is very handy for labeling macros for things such as flowcharting packages, and will play an important part in the dimensioning facility.

The centered text mode is "remembered" and if more text is entered by just pressing return at the next command prompt, subsequent lines will be centered under (in the logical sense, depending on orientation) the original line.

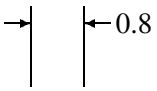
On 12/19/82 I completed the installation of "simple dimensioning". This is a form of automatic dimensioning derived from a description of how Computervision does it as described by Jamal Munshi. Some changes and additional features were added in this implementation. The facility is controlled by the DIM command. The entire facility is in an overlay, and if the feature is not purchased with the package, the overlay can simply be deleted from the release disc. COMMAND will catch the overlay not found error and print the message "Optional feature not installed." and return to the command prompt.[98]

[98] Dimensioning was the first new feature installed in AutoCAD after COMDEX 1982, and the first deliberately intended as an extra

First some background and terminology: a dimension consists of two "extension lines" (referred to as "witness lines" in some texts) which lead from the points being dimensioned to the specification of the dimension, a "dimension line" which has arrows on the end and which points to the extension lines, and the "dimension text", which gives the actual dimension between the two extension lines. If there is sufficient room between the extension lines, the dimension line will be within the extension lines, as in:



If there isn't room between the extension lines, the dimension line will be split and point to the extension lines from outside, as in:



In my implementation, dimension lines always run either horizontally or vertically. You can dimension a diagonal with a horizontal or vertical dimension line simply by running the extension lines to the ends of the diagonal line. To create a dimension, enter the DIM command. The first prompt to appear will be "First extension line origin:". You should designate the point near the item being dimensioned where you want the first extension line to appear. Next the prompt "Dimension line intersection:" will be given. You should then designate where the dimension line should be placed relative to the first extension line point (i.e., where the arrow will hit the first extension line). If the line between the first extension line origin and the dimension line intersection is more horizontal than vertical, the dimension line will run vertically, if more vertical, the dimension line will run horizontally. (Draw a couple and you'll understand.) Finally, you will be asked "Second extension line origin:" which requests you to designate the point near the object where the other extension line is to go. Before moving on to the rest of the command and its action, let me explain how these three points are used to define the dimension.

The first extension line will always start at the point designated for the first extension line origin. It will run either vertically or horizontally depending on the predominant direction between the origin point and the dimension line intersection. Any displacement of the dimension line intersection point from a true vertical or horizontal is ignored—it supplies only the direction and distance between the extension line end and the dimension line (or in other words, the dimension is taken from the extension line, not the dimension line, which is what you want). The second extension line is then drawn from the second extension line origin point, in the same direction as the first extension line, as far out as the first extension line went (note that the ends of the extension line at the dimension line always line up, but the ends nearest the item being dimensioned may appear anywhere, to accommodate strangely shaped objects). The dimension line will then be run either vertically or horizontally from the first extension line to the second.

There is a value called "arrow size" which influences many parts of dimensioning. The ANSI drafting standard specifies that drawn arrows should be 1/8 inch in size. Since we don't know the scale of the final drawing, we don't initially know how big to make the arrows. Thus we have a variable called arrow size which controls

cost add-on. The complete design of the initial dimensioning facility is presented here as an example of the developer documentation of the period. This very crude first cut at dimensioning became the entire content of the "Advanced Drafting Extension" (then called, "with Dimensioning") offered as a $500 option with AutoCAD 1.2 in April 1982.

this. If no arrow size is specified, we will guess and use 1/64'th of the smallest dimension of the drawing. The arrow size controls the size of the arrows themselves, which are drawn with solid fill at the ANSI-specified aspect ratio of 1 to 3. It also controls how far the extension lines extend past the dimension line (1 arrow length), how much spacing will be placed between the dimension text and the dimension lines (1 arrow length), and the initial size of the dimension text (1.5 arrow length). Further, if the arrows must be moved outside the extension lines, their shafts will have a length of one arrow length. The arrow size can be explicitly specified by the "DIM A" command, and will be saved in the drawing header. ANSI specifies that all arrows in a drawing shall have the same size.

Now we come to the matter of the text that specifies the dimension. The user will be prompted with "Dimension text:". If just a carriage return is given as the response, the distance in the direction of the dimension line will be measured and edited as a decimal number with four significant digits. Trailing zeroes and decimal points will be suppressed. If text is entered, the text will be used as-is. Since the text is terminated by a carriage return, spaces may be used within it.

Now that all of the components of the dimension have been acquired, the dimension is assembled on the screen. The first test is whether the dimension lines should be drawn inside or outside the extension lines. Our criterion for this decision is that if we have less than 8 times the arrow size between the lines, we'll draw the arrows outside, otherwise we'll try them inside. The text within the dimension is always drawn with angle 0 (that is, horizontally). This is in conformance with the ANSI standard, which specifies "unidirectional" dimensioning as the preferable technique. As a result, there are two cases depending on whether the dimension line runs horizontally or vertically. If it runs horizontally, we have to squeeze the text between the arrows so that the arrows run from the first and last characters of text. If it runs vertically, the lines run from the top and bottom of the letter area. In the horizontal case it's obvious that the length of the text is an important consideration in composing the dimension. In the vertical case we might just ignore the length (as only the height matters in fitting the arrows), but this isn't wise as we might overlay the object being dimensioned with the text. Consequently, for both the horizontal and vertical dimension line cases we compute a length in which the dimension text must be forced to fit; if horizontal, this is the total dimension line less 6 arrow lengths (2 for arrows, 2 for minimum length dimension lines, 2 for clearance between text and lines). If vertical, the fit length is two times the least distance between the dimension line and the extension line origin (this guarantees that the text won't overlap a straight figure; if the figure is convex, it may still overlap, and the user will have to move the dimension line further out).

Now that we know the space we have to work with for the text, we calculate the actual size needed to draw the text with its initial size of $1.5 \times Arrow_size$ (taken by measuring characters in ANSI-standard drawings). If the text fits at that size, it is drawn centered in the dimension line, and the lines extend to within 1 arrow size of the text. If the text doesn't fit, it will be reduced in size until it just fits in the space available. In the case where the arrows are drawn outside the extension lines and the dimension line runs horizontal, the text will be fit within 8 arrow lengths.

The extension lines and dimension line may have any geometrical relationship you desire; you may draw vertical dimensions top to bottom, bottom to top, and with the dimension line to the left or right of the extension line origins, and of course the equivalent for horizontal dimensions. The order you specify the ends makes a difference only when the arrows are drawn outside the extension lines, as the text will always be placed on the end of the second-specified dimension line (affording user control of placement), and when continuing dimensions.

At the point the "First extension line origin:" prompt appears, you may enter "B" or "C" to continue the last dimension. Either reply immediately advances you to the "Second extension line origin:" prompt, and draws

a new dimension based on the last one. "B" adds a dimension relative to the *first* point of the last dimension, drawing the dimension above (or whatever) the last one. This is what you do when you want a set of dimensions all relative to the same base line. Multiple "B" DIM commands will all reference the same base line. The "C" response draws a dimension relative to the second extension line of the last dimension, with the dimension line colinear with the last one drawn (unless the last one was outside the extension lines, in which case the new dimension line will be moved to clear it). This is what you do when you could care less about cumulative tolerances and want all the dimensions in a chain.

All the items that make up the dimension (lines, solids, text) are drawn as primitive entities, so they may be manipulated individually. The dimension facility is simply a tool to make these entities for the user, there is no special support in AutoCAD for dimensioning per se. The dimension code is split between DIM1 and DIM2 so as not to constitute the largest overlay and hence consume more memory—for logical purposes you can consider the two to be one piece of code.

On 12/24/82 I fixed a bug in ERASE reported by Jamal Munshi. This is one of the more humorous bugs I've seen so far in AutoCAD. Suppose you had a REPEAT/ENDREP loop with more than 1 entity in it. If you did an ERASE W and pointed to the entire bounds of the array elaboration, the scan of the refresh file would find each entity once for every array instance. Since there was more than one item in the array, the refresh file will contain a new entity header for each instance in the array. If the operation was an ERASE, and the total number of array items was even, ERASE which just inverts the entity type to delete the entity, would flip the sign an even number of times, leaving the entity not deleted. Since the UNDRAW was driven from the REDRAW file, the items would disappear on the screen, but come back on the next REGEN. I fixed this by making ERASE test whether the entity it's about to delete is already deleted. If so, it just leaves it alone. It's then also necessary to fix OOPS, since OOPS will also see duplicate entities and may re-delete something it shouldn't. Note that other entity designation commands can still be messed up by this case. I think that the real solution is to make EID scan the select list as it chooses each entity and guarantee that any given entity is selected only once. Maybe I'll change it to do that someday.

On 12/25/82 I installed support alternate forms of cursors, as available in the Aurora 1000 board, whose driver I am finishing up. The Aurora 1000 offers a software-selectable set of cursor types. I installed code in various places to support them, if available in a display device, in a device-independent manner. If the display device handles multiple cursor types, it should reference the external **`fixed(7)`** variable CURSEL. When DSMARK is called, if CURSEL is 0, the normal cursor should be displayed. If 1, then a box or window cursor should be displayed. If 2, a rubberband cursor should be displayed (i.e. crosshairs "pulling" a line). If CURSEL is nonzero, the other point (other box corner, origin of rubberband line) is given by the external fixed variables CURSLX and CURSLY. These variables are set to the screen coordinates of the last point returned normally by DIG—thus they are normally set to the origin point of the last point entered. Commands may set them (remember to use screen coordinates!) to other points if desired. Code installed to use this new feature is: EID uses a box cursor for the "W" form of designation; DSCMDS uses a box cursor for ZOOM W selection; EACQ1 uses a rubberband cursor for LINE acquisition, and EACQ2 uses a rubberband for TRACE acquisition. Note that if the display device doesn't support alternate cursor types, it may just ignore CURSEL, CURSLX, and CURSLY.[99]

I decided to go ahead and fix the multiple designation bug mentioned above correctly. I removed the kludge code installed yesterday in EDIT for the ERASE and OOPS commands. Then I installed code in EID which, for

[99]The Graphics Development Laboratory A-1000 (Aurora-1000) was the first true high performance graphics board AutoCAD supported. The A-1000 was a high resolution (512 × 480) 16 colour board with a high-level command set including true polygon fill, multiple cursor types, and window and viewport support on-board. Its capabilities encouraged implementation of congruent facilities in AutoCAD, and exerted an influence on the design of the package still felt today.

every point selected, scans the list of already selected entities and if it finds that the designated entity is already on the list, doesn't add it. Instead, it adds the entity to a duplicate selection list, DUPLST. This is necessary because if EID is going to delete the found entities from the refresh file (as for an ERASE or MOVE), it has to save all refresh file finds, even if for duplicate entities. After preparing the unique list and the duplicate list, if UNDRAW is requested, it undraws all refresh file entries in both lists. Otherwise, it just flushes the duplicate list and returns the unique list. This fixes all confusions known to exist when selecting entities within REPEAT/ENDREP loops.

On 1/1/83 I went back to work on the DataType 150 driver.[100] First some words about the design and history of this unfortunate device. It attempts to emulate a Tektronix 4010, but since its internal resolution is only 512×250, it translates the Tektronix coordinate space of 1024×780 into this raster matrix. This has a number of painful side effects. First, of course, the resolution is low to start with (vertical is the bad direction). Next, since we have to send Tektronix coordinates to the beast, AutoCAD believes that it has more resolution than is really available, so it draws things (text and circles) which aren't really visible on the screen. This makes things runs a lot slower than if the terminal were "honest". Since the Tektronix is a direct view storage tube, there were no commands defined in its command set for erase. Data Type extended the command set to include a dark vector, but did not include a complement vector. This has a horrible impact on the digitiser support, to be discussed below.

The DT150 driver, as currently implemented, uses the top 21 lines of alphanumeric space on the screen for the graphics display. The bottom 3 lines are used as a scrolling region for communication of commands. This is achieved by using the Televideo lock line command for the blank lines that overlay the graphics area, and leaving the bottom lines as a scrolling text area. Thus, the addressable graphics area is 1024×656. The terminal lacks any kind of area fill, so this is done with vectors as the AutoCAD default. Since there is no complement draw, implementing the screen cursor is basically impossible. I fixed it to draw a 3×3 (in Tek coordinates) cross. It draws this cross, then erases it whenever it moves and redraws it at the new location. Since there's no complement screen function, this means that the cross *erases whatever it passes over*! This means that as you use the digitiser, you slowly wipe the screen, until you're forced to do an explicit REDRAW to see where you are. I made the cursor small deliberately so that it wipes as little as possible.

Another offshoot of supporting terminals which "lie" about their resolution is that the BLIPs drawn for selected points may not look right. In the case of the DT150, they are so small that sometimes they look like "T"s. (Sometimes because it depends on where the center point maps into the modularity of the logical and physical resolution.) I made the variable in BLIP which determines the size of blip external and named it BLIPPO. A display driver is now free to redefine the size of the BLIP in its initialisation routine to be whatever displays correctly on the screen (n.b., should we change the closeness criterion in EID to equal BLIPPO? I think so, but I haven't done so).

In the process of working on the DT150 driver I discovered a "facet" of PL/I which I should warn other potential ploners about. Suppose you have a procedure which you want to pass different text strings for processing. You might declare that procedure as:

```
ZONK: PROCEDURE(PSTR);
      DECLARE PSTR CHARACTER(128) VARYING;
```

[100] In the early days of AutoCAD, we attempted to support some truly awful hardware in the hopes that its manufacturers would help us promote AutoCAD. To give you a flavour of some of the things we tried, I'm including a description of the attempt to get AutoCAD-80 running on the DataType 150, an RS-232 graphics terminal. The DataType 150 was neither the best nor the worst device we attempted to support. It shared with the best and the worst one common attribute: we never sold a single AutoCAD configured for it.

then call it as **ZONK('Booga')**. Be warned, the strong typing rules will force all the string constants you use as arguments to be 128 characters long, with the first as the length flag. Of course it will work fine, and the only way you'll notice you've been had is when you wonder where 30K of memory went. The most obvious way out is to make **PSTR** a global variable and assign the strings to it. Assignment doesn't force conformance on the right side, so the right side string will retain its attribute of **CHARACTER(*n*)**, and be converted to the variable string on the left hand side. This little change knocked 2K of wasted data space out of the DT150 driver, and I suspect that I can find similar savings throughout AutoCAD (in particular the PROMPT routine).

On 1/5/83 I "completed" the driver for the DataType 150. This terminal is such a pig that should we really wish to sell AutoCAD on it, a literally endless amount of work can be poured into its gullet without yielding a usable product. What I ended up with was a driver with no on-screen menu, a graphics area at the top, and a 3 line scrolling region at the bottom. Several days were wasted trying to get an on screen menu to work using the cursor positioning keys to make the menu selection. I abandoned this after it became clear that there were fatal flaws in the firmware in the terminal which precluded using this strategy. In order to implement the separate scrolling region and graphics area, I take advantage of the fact that the graphics and alphanumeric displays are separate and logically ORed to create the CRT drive. I use the Televideo lock line function to make the top 21 lines not scroll. Since they are cleared to blanks and don't scroll, they don't interfere with the graphics drawn in the same space. The bottom lines, being unlocked, scroll. I attempted to put a menu on the right side of the screen, but discovered that the cursor address function refuses to go to a line that's locked. It will only address lines in the scrolling region. To circumvent this, I tried unlocking the screen, updating the menu, then locking it again. Because of the way the line lock command works, this takes about 60 characters to do, so moving the cursor up and down the menu took an extremely long time to do. Furthermore, when running this sequence, the terminal would randomly get confused about its scrolling and insert blank lines in the menu area, transpose lines from the bottom 3 into the middle of the screen, and even scroll up the status line from the bottom of the screen. As this behaviour was completely random and not related to XON–XOFF handshaking what was previously checked out, I just decided to pitch the whole idea of an on screen menu and let this porker trot along without.

That left the issue of how to handle commands with lengthy output, such as STATUS and DBLIST. I installed a new procedure in the terminal driver, called DSLONG. (This was put in all the other drivers, but is null in them, as they are two screen versions.) Any command which generates lengthy output should call DSLONG(TRUE) before the first line of the output and DSLONG(FALSE) at the end. It must not exit without calling DSLONG(FALSE). The DT150 DSLONG procedure, on TRUE, clears the graphics and alphanumeric screens and unlocks the scrolling region so the whole screen can scroll. On FALSE, it displays the prompt "Press any key to continue:" and waits for an input character. When it gets one, it clears the screen, re-locks the graphics area, and calls REDRAW to put the picture back up. This kind of trick should work on any reasonable graphics terminal (I hope). The commands which activate DSLONG are STATUS, DBLIST, LAYER ?, and LIST.

On 1/7/83 I fixed some bugs in the shape compiler. First, SHCOMP was failing to test whether the shape number was between 0 and 255. Since the shape number is used to subscript a number of arrays, a bad shape number could lead to disaster. I installed an error message and made the compiler terminate with a syntax error. (Reported by Lars Moureau).

On 1/14/83 I installed code to aid in the use of devices whose support of colors is truly heroic (such as the Vectrix 384). The LAYER COLOR subcommand now accepts "*" as an argument. LAYER COLOR * will set all layers colors equal to their layer numbers (layer 1 color=1,... layer 127 color=127). Thus, drawings with

many colors may be created simply by assigning the layer number equal to the desired color.[101]

On 2/12/83 I completed support of the Hitachi "Tiger Tablet", HDG-1111. This is an RS-232 tablet very similar in general specifications to the HI-PAD from Houston Instrument. It offers either a 12 button cursor or a stylus as the digitising instrument. The driver, in file DSDRVTG, is very similar to the HI-PAD driver. The Tiger Tablet offers many protocol options, and the modes in which I chose to run it were chosen for similarity to the HI-PAD. Should we choose to recommend this device, we can switch to the more efficient and faster-to-track binary mode.

The stylus works like the cursor on the HI-PAD. Moving it on the tablet moves the crosshairs, and pressing it down selects a point. The 12 button cursor is implemented in the following way: moving the cursor on the pad moves the crosshairs. The zero button selects a point. Buttons 1 through 9 select the first 9 menu items immediately, regardless of the position of the cursor. Thus, very commonly used menu items may be set up for instant access through the cursor. Note that with either the stylus or the cursor, conventional menu selections may be made either by pointing to the screen menu and doing a point select (stylus press or zero button), or by making a point select within the digitiser menu.

On 2/21/83 I implemented a driver for the USI Optomouse.[102] This is a four button optical mouse which talks a protocol very close to the Summagraphics Bit Pad. The Optomouse driver (in file DGDRVOM) replaces the digitiser driver in AutoCAD. Assuming that the "tail" of the mouse is at the top, the buttons are numbered with the top row 2–1 and the bottom row 3–4. (The buttons are activated by pressing one end or the other of a bar on the mouse.) Button 1 is the "hit" button and selects a pointed to menu item or a point on the screen. Buttons 2 through 4 select the first 3 menu items regardless of the position of the mouse, and thus may be used for very frequently used commands, obviating the need to move the mouse to point at the main menu. (Note: if this doesn't work for you, it's because whoever generated ACAD forgot to turn on the KEYPAD compile time variable in DIG—just a word to the perplexed.) The Optomouse can be configured by output commands for various modes. Currently, the only configuration done is to program it for 10 ms between samples in order to speed up sampling rate and smooth cursor motion. Note that this driver was developed and tested with a prototype Optomouse and will have to be modified when the production mice with auto-baud rate sensing are delivered. Note that the mouse driver differs from a standard digitiser driver in that DGACOR isn't enabled, and that the output coordinates are clipped to prevent returning a sample in the digitiser menu area, which we can't use with a relative positioning device like a mouse.

On 2/23/83 I blind-implemented a driver for the Summagraphics Bit Pad to solve a customer requirement.[103] The driver was created by modifying the Optomouse driver and tested with the Optomouse. It differs from the Optomouse driver only in that it doesn't send the command string to send sample rate, it turns on DGACOR, and enables the digitiser menu as for the Hi-Pad and Tiger Tablet. The driver was tested with the Optomouse and works, but customer comment will be required before the driver can be considered operational.

On 3/11/83 I completed a complete rewrite of the plotter driver, encompassing both the device-independent part and the Houston Instrument driver. The rewritten driver is now compatible with the version in AutoCAD-86, the offshoot of the Plotter Integrity Project. The changes between the old and new drivers are massive, and will be discussed in something approaching order of importance.

First, there is now a plot configuration file which saves all the parameters for the plot. **ACADPLOT.CFG** is

[101] This was the first device we supported with a large colour gamut. It supported 512 colours from a palette of 16 million. It was five years ahead of its time.

[102] This was the first mouse ever supported by AutoCAD.

[103] QA? What's that?

written and updated by the plot driver, being created the first time a plot is made. It contains the plotter manufacturer name, the plot paper size (either a standard ANSI or DIN size, or a custom size specified by the user in inches or millimeters) the units the plot is in (again Inches or Millimeters, also selecting ANSI or DIN sheet size nomenclature), and the pen width (of the narrowest pen), which is used to calculate the number of vectors needed to FILL a solid and the accuracy required for true circles and arcs.

The file also contains a driver specific configuration section. For the Houston Instrument driver, this specifies the model number (DMP-?), the assignments of pens, line types, and velocities to each color in the drawing.

The plotter driver is now organised so that one master driver can be supplied which runs all plotters we support. The code has been written so that drivers for each manufacturer can be overlays, so there will be no limit on the number of different plotter protocols we can support in one driver.

Once the standards for a plot are specified, they need not be entered again unless the user wants to change the standards. If a change is desired, the user can go through the items, changing only those desired. A null entry to any configuration query leaves the old value unchanged. An interactive entry dialogue is provided in the HI driver to set up the assignment to drawing colors. "X" ends the dialogue. If the parameters are changed, they may be used in the changed form for just this plot or for all future plots, at the user's option.

Handling of scale and windowing the plot onto the paper is totally different. If the SCALE query is answered with a null reply, what is plotted will be identical to the screen contents, nothing more and nothing less. The paper size will be adjusted automatically to accomplish this (and a message will inform the user of the adjustment). Scale may now be answered with a number (specifying a scale greater than 1 to 1 (optionally followed by an "X")), or by a scale less than 1 in the form "1:n" or "1/n". This scale will map the drawing units to physical units on the paper according to the units in effect. If the plot is in English units, a scale of 1 will map 1 drawing unit to 1 inch. If metric, 1 drawing unit will become 1 millimeter on the paper. A scale of 2 will make things twice as big on the paper, and a scale of 1:10 or 1/10 will make things ten times smaller.

All input to the plot driver is now handled with a data input routine like that in the drawing editor. Thus space, CR, or ";" may now be used to terminate input, and ↑H and ↑X work as local editing keys normally. ↑C will now abort the plot during configuration, not return the user back to CP/M unexpectedly.

The Houston Instrument driver supports only 15 logical colors (1 to 15). Colors greater than 15 will be wrapped around modulo 15 (e.g., color 16 becomes 1, 17 becomes 2, etc.). This corrects a previous bug where out of range colors would result in the ridiculous action of faithfully drawing with no pen.

On 3/13/83 I made a large number of changes in the main menu module, **AUTOCAD**. If ACAD is serialised with dealer number 97, our code for an evaluation version, the main menu will now print "**EVALUATION VERSION---NOT FOR SALE**" in the drawing header for all displays.

I fixed a bug in ZOOM W. A null or line window would cause ACAD to ZOOM into hyperspace. I fixed it to reject a window without area. (Reported by Greg Lutz).

Aaaaah yes, the HI-Pad.... Well, the scum also rises, and it's time to resolve the issue of Hi-Pad jitter once and for all. I spent about 2 hours looking at the character stream sent by the HI-Pad as I subjected it to various "stimuli" such as static electricity (thanks to a cooperative (?) cat), magnets, and proximity to the RGB designs "Big Mutha" monitor and a Hazeltine terminal. Except in cases of extreme and unreasonable abuse (25 KV kitty dragged across the pad), the jitter consisted of a random ± 10 in the Hi-PAD sample. This jitter was completely random, exhibiting no periodicity or pattern that old "Random" Walker could detect. Run lengths varied randomly from 1 to 100! There is *no* smoothing technique which can compensate for this. The only

answer is resolution reduction. So I fixed DGDRVHI to check the sample about to be returned against the last sample returned. If the absolute value of the difference between the current sample and the last sample is 10 or less, the last sample is returned. This results in a pad which is highly responsive to fast moves (as the JITTER averaging code can be disabled), but which settles to rock-steady stability when the cursor stops. But where's the catch? Well, it's very simple. Ignoring ± 10 excursions from the pad makes only movements of 20 points or more meaningful. With a pad resolution of 11000 in X and Y, this means that the pad can't address every pixel in a display with more than 512 dots in any direction. But then, with its jitter it would never succeed if it were to try! So, the HI-Pad has been bashed into acceptable performance for those applications for which it is usable.

I further changed the HI-Pad driver to automatically re-sync if synchronisation has been lost with the data stream. This prevents cursor flashing if an exogenous delay causes loss of sync with the data from the pad (such as a long cursor draw time).

I also changed the Hi-Pad driver to lock its output coordinates within the 0 to 22000 limits by forcing out of range coordinates to the extreme limits rather than rejecting samples outside the range. This makes it much easier to select menu items and points near the edges of the screen. This mode is clearly preferable and will be the standard for all digitiser drivers in the future.

On 3/15/83 I hunted down the "8080/8085" bug and installed a workaround in ACAD. For certain values of X, Y and Z, the expression: $(X + Y) < Z$ yields incorrect logical values on an 8080 or 8085 processor, but the correct value on a Z-80 processor. In general, it appears that the use of an expression as an argument to a relational operator can lead to the ambiguity. This caused circles, arcs, text, and traces/solids not to generate in certain cases. The basic bug is in PL/I-80 and an 8 line program was prepared which gets different answers on a Z-80 and 8085. A bug report was submitted to Digital Research. By "Captain Empirical" techniques, it was determined that assigning the expression part of the relational to a temporary variable eliminated the error in evaluation of the relational expression. The code which used such expressions for the off-screen tests in EREGEN (ARC, CIRCLE, and TEXT), and FILL (TRACE, SOLID) was rewritten to bypass the bug until DR supplies a fix.

On 3/23/83 I was foolish enough to try to implement a driver for the Mouse Systems M-1 mouse. I'm not sure if "mouse" is the correct designation for the position of this little bugger within Rodentia. The main problem is that, unlike the USI mouse, the MSC mouse only sends data to the host when something changes. This means that motion of the mouse when the computer is off doing something else is lost, since the samples just pile up in the UART and get lost. Worse, the sample format is 5 bytes per sample, without parity, and with no unique synchronisation code for the first byte (it's a pattern "unlikely" to be used in the 4 binary data bytes which follow). So, you can fix it to time out when out of sync and know that it will re-sync on the next sample, and you can put up with not being able to move the cursor except when the computer is tracking, because that's the way most mice work anyway. But, now for the catch: when you push a button, if the mouse isn't moving, it sends just one sample on the push and one on the release. If you happen to miss the push because you're out of sync or off updating the cursor or checking the console, then the button push is lost forever. I consider this device only to be useful if supported with an interrupt driven driver. Since that's not practical in AutoCAD-80 with its universal configuration requirements, I don't consider this mouse to be usable. And it's a damn pity. The thing has a microcomputer in it, and could have been programmed to work reasonably.[104]

[104]The Mouse Systems mouse was an interesting device. To my knowledge, this is the first peripheral we encountered which was designed assuming the IBM PC as its primary host machine. CP/M machines usually didn't run their serial ports on interrupts, but IBM PC's did, and the Mouse Systems mouse lived long and prospered in that environment. I am editing this document on a Sun workstation equipped with a Mouse Systems mouse.

On 3/25/83 I implemented a driver for the Strobe model 100 plotter. This is a small A size plotter which talks a subset of the Hewlett-Packard protocol. The driver was added to the universal plot driver and is selected by answering STROBE to the manufacturer query. As this is a single pen plotter with no dashed line or variable speed support, there is no plotter-specific configuration at all. The plotter works magnificently, and no programming nor mechanical quirks were encountered in bringing it up or developing the driver for it. Since this is an RS-232 device, the port is configured in INSTALL exactly as for the Houston Instrument plotter.[105]

I also took Richard Handyside's HP driver and created a universal HP driver and installed it in the plotter driver. I suspect that it will do something vaguely reasonable, but won't work right until I understand a bit more about the family of HP plotters. It is accessed with the HP manufacturer specification, and has a configuration dialogue almost identical to the HI plotters. Configuration at the INSTALL level is identical, as the HP plotters are RS-232.[106]

On 4/1/83 I installed a CHANGE LAYER command. This command works on any entity type, and allows objects to be easily moved from layer to layer. To use it, give the CHANGE command. The usual prompt to select objects will be given. Then designate the objects to be moved between layers by any of the standard methods. When the prompt "New location" is given, answer "LAYER" (actually, anything beginning with "L" is OK). The prompt "New layer:" will then be issued, and you should respond with the new layer number. The selected objects will then be redrawn on the new layer (if visible).

I extended the CHANGE command to work on TEXT. The location, size, angle, and content of text may now be modified by CHANGE. A null response to any of the queries will cause the selected modification to be ignored.

On 4/19/83 I installed code to allow a demo version of AutoCAD-80 to be generated. The demo version is identical to the regular version, except that it will not write out an output file, nor will it make a plot from within the drawing editor (plots from the main menu are OK). This results in the user being able to try all commands on the sample drawings supplied with the package, and to make new drawings, but since no output may be saved on disc, no real work can be done.[107]

On 1/12/84 I installed a gimmick in DSDRVAUR to provide support for the A-1030 colour mapping board. This option provides a user-loadable palette which can map any of the 16 color numbers generated by the A-1000 into any arbitrary analogue RGB value. Since if this board is present, the colour assignments are up to the user, we do not want AutoCAD's normal colour translation before the codes are sent to the A-1000. The configuration variable LPUSED, previously not used for the A-1000 (it is the light pen present flag for the Scion), is used as the A-1030 present flag. If zero (the default value if not specified) the A-1030 is assumed not to be present and colours are translated into the standard values as before. If LPUSED is set to **`OFF`** in the **`.INP`** file, then the A-1030 will be assumed to be configured and colours will be sent to the A-1000 without translation (modulo 16, of course). Thus, the user can set up any mapping he wishes by loading the translation tables in the A-1030 and AutoCAD will not stand in the way.[108]

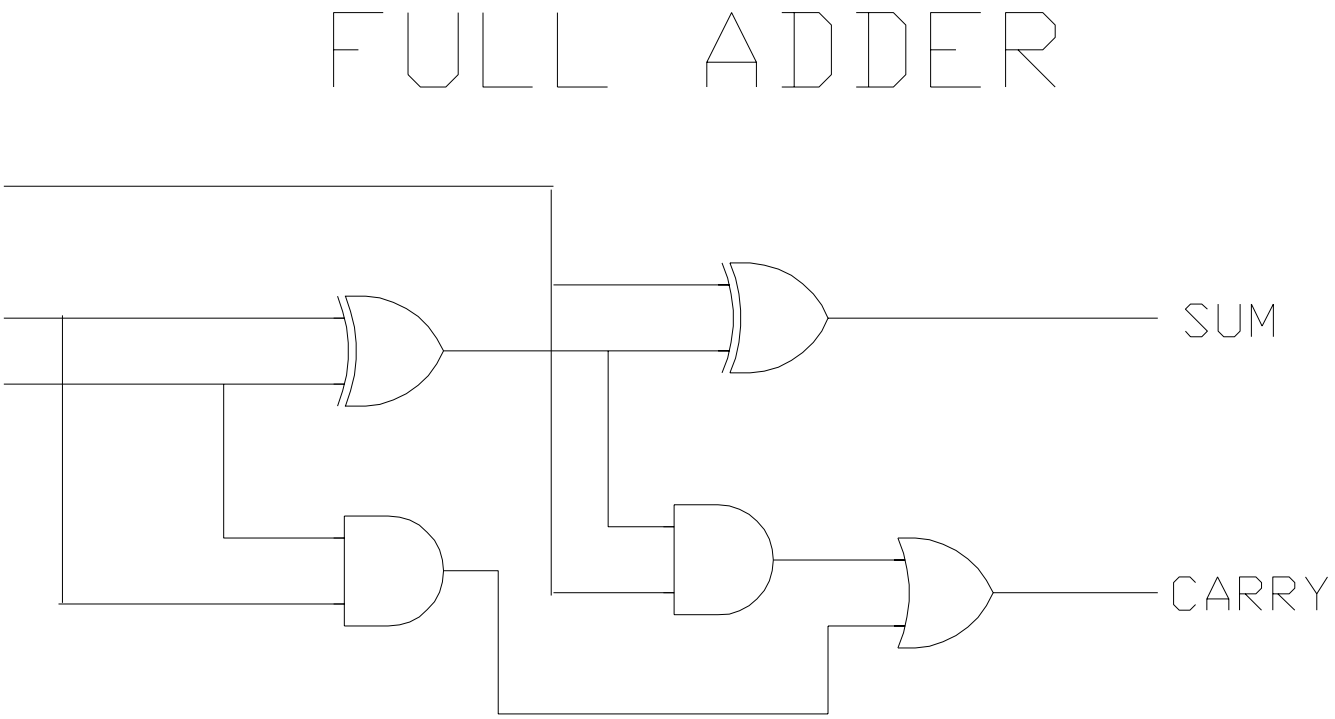
[105]This was a remarkable device. You wrapped the glossy paper around a drum and screwed the nib end of a Pilot razor point pen into an 8-32 nut brazed onto the pen-down actuator, and away it went. It produced such good plots that we used it to make the illustrations for the first AutoCAD manual we printed.

[106]Hewlett-Packard plotters are supported. Up to this point, we only drove Houston Instrument plotters.

[107]Nobody ever imagined that there would be a need for a demo version of AutoCAD-80. But there was, anyway.

[108]This was the last work done on AutoCAD-80. Additional copies of the final product were sold in 1984, but the development of the product ceased and effort was focused entirely on AutoCAD-86.

This electronic schematic drawing and all of its constituents, appeared on the original sample drawings disc. It was the first sample drawing originated on AutoCAD-86.



1982 Annual Meeting

With the push to get AutoCAD running underway, and other Autodesk development projects proceeding concurrently, there wasn't much time to write minutes of meetings, though the meetings continued to be held every month. The next general communication was the announcement of Autodesk's first annual stockholders' meeting. Dan Drake alludes to some of the overwork and the communications problems this engendered herein.

Autodesk: Monthly Meeting

Dan Drake
December 28, 1982

Our good old algorithm for determining monthly meeting dates strikes again: the January meeting would come up on the first Saturday of January, alias New Year's Day. Of course, we don't have it on a holiday, so it moves up a week, to December 25. Oops.

So. . . .

The January monthly meeting will be at 1:00 on Saturday, January 8 (*eight*), 1983, at Jack Stuppin's house in San Francisco. At 2:00 it will be interrupted for the annual shareholders' meeting (required by our bylaws and state law). This probably won't take long, as the only item currently on the agenda is the election of directors (required by law). Then the directors will meet, again required by law and again probably very briefly, to organize the board and appoint the corporate officers. With luck, if no new business comes up, we may break the world speed record for a bona fide annual meeting.

Those of you who are now stockholders of record will receive a formal notice and proxy with this announcement. If you exercised an option by the 28th, you're a stockholder even though you haven't got your certificate yet.

Employees who aren't stockholders, and stockholders who aren't employees, are encouraged to attend the whole shebang.

The fact that this is the first written notice in months brings up a sore point. With respect to communications in the company, people have taken two positions:

- Where the hell are the information letters; we haven't had one in months; nobody knows what's going on; we can't survive without better communications.

- Putting out minutes or an information letter knocks a couple of days out of the time of someone who has to be getting a product ready for market; we can't survive without getting products out.

Both of these positions seem to be quite right. Anyone who has constructive proposals on the subject is guaranteed a high place on the agenda for the meeting, if any of us has time to prepare an agenda.

Finally, here's the meeting algorithm again, with its output for the coming year. Should this silly thing be changed?

> The meeting for an odd-numbered month is on a Saturday; for an even month, on a Sunday.
>
> The meeting is normally on the first (Saturday or Sunday) of the month.
>
> If that day falls on a holiday weekend, the meeting is held a week early. This is certain to happen in July and September, likely in January, and possible even in April.

The schedule for the rest of the year:

Saturday, January 8
Sunday, February 6
Saturday, March 5
Sunday, March 27 (April 3 is Easter)
Saturday, May 7
Sunday, June 5
Saturday, June 25
Sunday, August 7
Saturday, August 27
Sunday, October 1
Saturday, November 5
Sunday, December 4?
Wednesday evening, December 7 (annual meeting)

December 29, 1982
NOTICE OF ANNUAL MEETING

The annual shareholders' meeting of Autodesk, Inc. will be held at 2:00 PM on Saturday, January 8, 1983, at the home of Jack Stuppin in San Francisco.

(The meeting was re-scheduled by order of John Walker, president, from the normal time of the first Wednesday in December, when most of the company was in Las Vegas.)

Stockholders of record December 28, 1982, will be eligible to vote at the meeting. If you are not absolutely sure that you will attend, please sign and return the enclosed proxy, which will be revoked automatically if you do attend. It looks better on the records if we do not drag through with a bare quorum.

The only business currently scheduled for the meeting is the election of four directors for the next year.

Daniel Drake

Daniel Drake
Secretary

PROXY

I appoint JOHN WALKER, DANIEL DRAKE, and KEITH B. MARCELIUS my proxies to vote on my behalf at the annual shareholders' meeting of Autodesk, Inc., to be held on January 8, 1983, and at any adjournments thereof. All previous proxies are hereby revoked. The majority of said proxies present at the meeting, or their appointed substitutes, shall have the power to vote my shares in the election of directors or any other business which may come before the meeting, with the following restrictions:

I may revoke this proxy (1) by giving written notice to the corporation; (2) by executing another proxy, which will take effect when presented to the corporation or exercised at the meeting; (3) by attending the meeting and voting in person.

(signed) _____ *Name* _____ *Date* _____

Quality Department Priorities

Even as we were shipping the first AutoCADs, we were concerned with achieving and maintaining the highest quality standards for our products. Since our money and marketing resources were so limited, we had to rely on our reputation as the primary sales tool. Our resources also prevented us from actually starting a formal Quality department until Mauri Laitinen started full-time to undertake that task on January 4, 1984, precisely one year later.

Quality Department Priorities

by John Walker — January 5, 1983

One of the central but almost unspoken assumptions of AI since its inception is that we would supply products of superior quality. In the software business, this is the key to survival, growth, and respect. Nobody in this company is going to say "let's ship this shoddy junk", but as we grow and have to support more features, more machines, and more peripherals, it becomes harder and harder to verify that every product we ship meets our intended standards of performance. Thus, in order to keep our performance in line with our intentions, we now need a Quality department which will help us achieve our goals. The Quality department is not responsible for the quality of our product—it is responsible for developing the procedures which verify that the company as a whole delivers only products which meets our standards of performance.

The following are what seem to me to be chief priorities in the establishment of a coherent quality function within Autodesk.[109] All items listed below are of equal priority unless otherwise noted.

Development of Specifications

Quality is adherence to specifications. What are the specifications for AutoCAD? One straightforward approach is to develop a test suite which will execute on an AutoCAD which meets the specifications embodied in the manual.[110] This test suite and ancillary tests should become one of the prime validation tools used to qualify a release from the Technical department for release.

[109] This entire proposal is based on the principles enunciated in *Quality Is Free* by Philip B. Crosby, Mentor, 1979.

[110] Good idea. Somebody should get around to that some day.

Qualification Testing

The Quality department should conduct all initial testing of new proposed releases by the Technical department. This testing should consist of testing against the specifications of the previous release, and development of tests for new features and changes in the new release and their integration into the test set.

Test Management

Once a new software version has been proposed for release, Quality shall cause it to be shipped to beta test sites, and shall coordinate all test responses and problem resolution.

Discrepancy Logging

All reported discrepancies (e.g., bugs) should be logged by Quality and tracked to resolution. Quality will prepare and present to management summaries of problems encountered and resolved on a regular basis. Discrepancy reports will be retired only upon the certification by Quality that the problem has been corrected. Quality will develop reporting with the cooperation of Customer Support and Technical groups to implement this function.

Release Certification

Quality will certify all software for release. Operations will be provided with master copies of software by Quality after this certification is complete. Operations will ship no software not certified by Quality.

Regression Testing

Quality will develop and implement procedures to verify that problems reported and resolved do not reoccur in subsequent releases. Development of a regression test suite and monitoring of Technical source code control procedures are among the tools which may achieve this goal.

Cost of Quality

Quality shall prepare and present, on a monthly basis, reports to management on the direct costs incurred by AI as a result of quality problems. These reports shall include customer support time, product replacement costs, lost production time, and any other costs traceable to shipment to customers of product failing to meet AI specifications.

Meetings: December ’82, January ’83

This document chronicles the period when Autodesk turned the corner. Before we went to COMDEX in 1982, we were a group of idealistic programmers with a dream. We returned from COMDEX with a product that we knew to have demonstrated its potential to be a mega-hit. The problem then was how to best deploy our meager resources to best exploit the opportunity our work had created. In the December and January meetings we began to grope our way toward approaches to the challenges that faced us.

December and January Meetings

by Dan Drake

This is a quick summary, long after the fact, of the general meetings of December, 1982, and January, 1983. The keeping of proper minutes, which was abandoned for months because of the lack of anyone to do it, will be resumed as of the February meeting.

December Meeting

The December meeting was held on Sunday, December 5, at Jack Stuppin’s house. The major topic of discussion was COMDEX, which had been held the previous Monday through Thursday.[111]

Our booth at COMDEX had AutoCAD on the Z80, Victor 9000, and IBM PC. We were also passing out brochures on Autoscreen, but made no real effort to demonstrate it, as an editor doesn’t make a very gripping demo.

AutoCAD was on display in a total of four booths: Sierra Data Systems, SunFlex, and Victor, in addition to our own. With the aid of all this exposure, our obscure booth in the back of the show was almost continuously full of people—to the extent that Steve Ciarcia couldn’t find his way in. Win a few, lose a few.

It was apparent that AutoCAD was a hot product. At the biggest show in the industry, it had the field to itself, with no direct competition at all. The main problem before the meeting was how to cope with success.

Mike Ford pointed out that exploiting the large number of leads we brought back would be a full-time job. Being unable to work full time for any extended period with no pay, he suggested that Autodesk pay him a

[111] This, of course, was the COMDEX at which we introduced AutoCAD to the world.

10% commission on sales, up to some reasonable amount, for a few months. There was much discussion of the advisability of paying someone at this point, of whether we should pay ourselves commissions, and of the relation between taking a cut on sales and taking a piece of the company's growth by means of stock options.

There was a consensus that paying Mike would be a good investment, and that the commission arrangement would amount to paying him a salary with him taking all the risk in case we didn't get enough income. The following arrangement was approved by unanimous vote:

Starting immediately, Mike gets a 10% commission, to a maximum of $6,000 a month (a figure intended to compensate for the risk that he'd get very little, and corresponding to nearly 3/4 of a million a year in sales). The arrangement runs for three months, and can be renewed by the directors month by month. When it runs out, the commission drops to 8% for a month, then 6%, and on down to zero. The idea behind the gradual decrease was that some deals, including the most profitable ones, give delayed yields.

January Meeting

The meeting on Saturday, January 8, was the normal general meeting with the annual shareholders' meeting and board of directors' meeting sandwiched in.

The shareholders' meeting elected an almost unchanged slate of directors: Dan Drake, Mike Ford, Keith Marcelius, and John Walker. Jack Stuppin declined to continue to serve on the board because of problems with stock exchange regulations, but will work unofficially with the board.

The board in its turn confirmed the corporate officers:

John Walker	President
Daniel Drake	Vice President, Secretary
Keith B. Marcelius	Treasurer, Assistant Secretary
Robert R. Tufts	Assistant Secretary
Roxie Walker	Assistant Treasurer

The reason for all the funny assistants is that the signatures of one or more officers are required on many formal documents, including the application for a California resale permit. With an assistant treasurer in Marin and assistant secretaries in the Eastbay and at the law office, we save a good deal of time and hassle.

John Walker presented the financial report, the crux of which is that our liquid capital is down to about $17,000.[112]

We have two sources of additional capital. Marinchip has warrants to buy 40,000 shares at $1.00 a share; it bought warrants because it lacked the cash to buy in for that much last spring, Marinchip now has enough cash to exercise part or all of the warrants. Also, the original capitalization plan included 20,000 options for John Walker; by oversight, these haven't been awarded yet, but the board intends to approve them as soon as it can print the paperwork.

Sales up to January 8: 4 AutoCAD-80; 1 AutoCAD-86; backlog of 2 AutoCAD-80s; 2 AutoScreens. AutoScreen is now part of the Engineering Work Station sold by Jamal Munshi of MOMS Computing.

[112]This was the low point. From there on, it was up, up, and away.

AutoCAD-80 is running on the Aurora 1000 from Graphic Development Lab, which has sent out 500 of our brochures; the product will be officially released soon. A dimensioning package is being released as an independent add-on for $500 list.[113]

AutoCAD-86 is scheduled for release on both the IBM PC and the Victor/Sirius on January 15.[114] The features included will be those in the preliminary AutoCAD manual, with an upgrade somewhat later. Sun-Flex, the maker of the Touch Pen, will be selling their product with AutoCAD (for the Victor) and is talking of selling 100 in February, gearing up to 300 a month soon.

We have many contacts with companies that are interested in AutoCAD, including Fortune and Houston Instruments (which wants a demo of the things we can do with their plotters).

Jack Stuppin has connections in the educational publishing business, which could lead to some profitable ventures. It is not too early to think about what programs we'll produce after AutoDesk has been released and AutoCAD has stabilized.[115]

Dave Kalish raised the idea of a sort of personal computer software division, built around people in the South Bay who are unable to work on AutoCAD because of conflict of interest and geographical isolation. There was general agreement that this was a good idea. Possible projects include a menu-planning package that Dave is working on with some other people, and the sort of educational programs that Jack's associates might put us onto.

To improve communications in the company, there will be a bi-weekly mailing of interesting items to everyone in the company. Interesting items might be anything from news clippings to suggestions for ADI products to project reports. Announcements and minutes will be in the package if they come out at the right time. Items sent to Roxie at headquarters will be Xeroxed and sent with the next mailing.

There was a strong consensus that we must have minutes of meetings in some form, and an agenda enforced by someone with a mandate to keep meetings from rambling. Dan Drake agreed to resume the minutes and assume the role of enforcer if no one else volunteered, and was elected by a unanimous silence.

Duff Kurland expressed interest in working full time, if some kind of arrangement for pay could be worked out so that he would not be in effect plowing a few thousand a month into the company. This brought up the situation of Greg Lutz, who has been working full time on AutoCAD-86, and Dan Drake, who has given ADI preemptive priority over paying work; with both having to switch into a non-destructive financial operating mode, the project's technical manpower was about to be reduced to one part-time person.

By unanimous vote it was agreed that Greg would get a subsistence of $1,000 a month for the next four months, in return for which he could give AutoCAD-86 his undivided attention.[116] Everyone else was asked to submit a letter to Jack Stuppin, giving the specific conditions under which he would be able to work full time for ADI. With those letters we could plan to take on more full-time people as the cash flow picked up.

Last October or thereabouts there was to have been a special all-day brainstorming meeting on how best to manage the company, but it was bumped by one crisis or another. Dave Kalish pointed out that there was as much need as ever for this meeting, so it was rescheduled for Sunday, February 13, tentatively at Jack Stuppin's house at 11 A.M.

---

[113]Thus began the pattern of extra-cost options. Later, dimensioning was combined with other features and sold as "ADE-1".

[114]That's right, AutoCAD-80 was shipped in December 1982, but AutoCAD-86 wasn't shipped until January of 1983.

[115]Who knows what we will do once "AutoCAD has stabilized".

[116]Thus, Greg Lutz became Autodesk's first full-time employee, at least in terms of salary.

One thing that has become apparent is that we often don't know, or don't agree on, what has been actually decided at a meeting. Here, then, is the first instance of a feature that will appear in all minutes, though perhaps in some prettier form. If you think it's inaccurate, speak now or forever hold your peace.

Summary of Decisions Taken

Mike Ford will temporarily get a commission on sales, described in detail in the DECEMBER section.

Greg Lutz is on the payroll at $1,000 a month, effective January 8.

The board of directors is the same as last year, except that Mike Ford replaces Jack Stuppin. The officers are the same.

There will be minutes and an agenda for the meetings. If you have something that needs to be discussed, inform Dan Drake before Friday night, or be prepared to be gavelled down.

We will meet on Sunday, February 13 to figure out how to run the company. This is to be an all-day session, though we can't start it before 11:00 because of travel time. Bring your specific complaints, your general principles for running things, your specific ideas, and a readiness to volunteer for the grungework of carrying out the ideas.

February 1983 Meeting

The February 1983 meeting looked back on not only COMDEX, but the second and third trade shows we attended: CADCON and CPM-83. The strains of converting a loose aggregation of entrepreneurially-oriented programmers into a responsive company can be seen in Dan Drake's announcement of the meeting. A large quantity of collateral communication led to the feeling that we had to resolve the issue of effective management.

In February of 1983, the company's cash position turned around and, in fact, doubled in one month. AutoCAD sales had begun to take off, and began a growth curve which, at this writing four and a half years later, has not yet seen its first inflection point.

Autodesk Monthly Meeting
February, 1983

Dan Drake

The February monthly meeting will be at DUFF KURLAND'S meeting room (not Jack's) at 1:00 on February 6.

On February 13 we'll have the special meeting on managing the company. To approximate an all-day meeting, this is starting at 11:00 (*eleven*). It's currently scheduled for Jack's house. When originally proposed, way back in October, this was described as a brainstorming session; i.e., lots of ideas should be put forth before we start inhibiting ourselves by looking at them critically and realistically.

[Unauthorized editorial by DD: This is not a company owned and operated by some guy or guys in the Northbay. It is a cooperative venture of some entrepreneurs, all of whom think and act as if it were a company owned and operated by some guy or guys in the Northbay. This fallacy is self-sustaining and is very close to destroying the company.

[There are two equal and opposite errors concerning how to fix things. One is to exhort everyone to buckle down, work hard, and act like an entrepreneur. The other is to figure out better management strategies for the people who run things to use in managing the workers. Can a person who's miles out of the daily action, able to commit only a small amount of time to the enterprise, really be blamed for not feeling much like an entrepreneur? Will he feel better if there's a better bureaucracy, more like the conventional company he works for (considering that he's not so devoted to that company as to devote all his time to it and resist new things like ADI[117])? Are there any practical, concrete steps by which we can discipline or trick ourselves to see the

[117] ADI here was used as an abbreviation for the company name. In 1985, it came to mean "Autodesk Device Interface".

company correctly? Not that this will solve all the management problems, but nothing else will without this.

[Finally, if you come to the meeting with a proposal that somebody should tie a bell on the cat's tail to warn us when it's coming, don't forget to consult your Boy Scout Handbook (to review knots) and your family attorney (to update your will). There's no use for proposals of the form, Let's You And Him Fix It This Way.]

Responsible opposing viewpoints will be given equal time; that is, if you have an idea you want circulated before the meeting, get it to me in writing (mail, MJK, courier, or whatever) by Tuesday Feb. 8, and I'll mail it to everyone on Wednesday.

February General Meeting

by Dan Drake

The February general meeting was held on Sunday, February 6, at Duff Kurland's. Present were Dan Drake, Mike Ford, Dave Kalish, Duff Kurland, Greg Lutz, Mauri Laitinen, Keith Marcelius, Hal Royaltey, John Walker, and Roxie Walker.

Minutes and Financial Report

There was no dissent from the minutes of the December and January meetings.

John Walker presented the financial report. As of January 30, one year after the first organizational meeting, the company had a positive cash flow, with net assets doubled from the previous month.[118]

Bank account:	7,966
Capital Preservation Fund	17,987
Liquid assets	25,953
Receivables	7,734
Total short-term assets	33,687
Liabilities & committed exp.	(1,500)
Net	32,000+

Expenditures in 1982 amounted to $44,493, broken down as follows: hardware 48%; shows 22%; legal expenses 14%; printing 7%; stock repurchase 5%; others 4%.

Board of Directors

The Board is now meeting informally during the week before each general meeting. At the meeting on February 2 the main topics discussed were marketing plans and manpower allocation.

The Board's idea, discussed and amplified at the general meeting, is that we should offer a standard distributor and OEM price of $425 or $450, and let them tell us what they can offer in return for the still lower price that they'll ask for. When people have new machines that they want us to put AutoCAD on, we'll try to negotiate an engineering fee for the work. If an OEM looks valuable, we may convert to his machine on speculation (as we have done up to now) and make an evaluation copy for him to look at, but without committing ourselves to anything in terms of an eventual release. Any manufacturer who is serious will give us a loan of his equipment for as long as we're supporting it in our software.

[118] The corner had been turned. From this point on, Autodesk was profitable.

As to manpower, the Board noted that there would be no manpower for marketing AutoDesk if it were ready for release immediately; our marketing operation is over-loaded trying to get full advantage out of AutoCAD. The implications of this were discussed later.

Project Reports

Marketing

We are pursuing listings for AutoCAD in every relevant directory and are beginning to get announcements published. An article in ISO World got about 200 responses. We are sending announcements to about eighty industry analysts and a couple of hundred OEMs. An important goal now is to get a number of retail dealers around the country, who can answer questions and give demos. Of course, we went to two trade shows in January.

There was a longish discussion of where we should spend money in promoting the product, particularly whether we should buy into a mailing of card decks by the *S. Kline Newsletter*. For instance, if we burn most of the card decks we get, why think that everyone else won't burn theirs? The upshot was that we should budget $1,000 for Mike Ford to try out what he wants. It will take a month or two before we have enough cash-flow information to allocate a proper budget for advertising.[119]

One of our objectives is not to be the Electric Pencil of the CAD industry (90% market share in word processing in 1978; zero in 1982). To do this we want to nail down as many manufacturers as possible, so that the next product to come along will find very little market left.

We are working with USI, which makes a cheap ($300) mouse and wants to use AutoCAD to push their product. We have sales now at Ford and Shell in England, with prospects for very large sales in the future. Houston Instrument plans to use AutoCAD to demonstrate its products at shows. Texas Instruments has an evaluation copy and is of course going to require 10,000 AutoCADs on its new IBM clone.

AutoCAD-86

Dan Drake is now the interim sub-project manager for AutoCAD-86 (until somebody complains about the cooking[120]). A large part of the job consists of talking to the outside world. Technical people on the project (meaning Greg Lutz at the moment) have unlisted phone numbers as far as the outside world is concerned, and should be called as little as possible by our own people.

AutoCAD-86 release 1.1 went out around January 15 after a struggle with the document. There is a demo version, which is a full program except that it can't write any output on disk; it will also be fixed so that it can't plot from within the drawing editor. We solicit any suggestions for how people could get around the limitations to get useful work out of the demo version!

The next release is scheduled for March 10, with the full features of AutoCAD-80, plus the Epson screen dump on the Victor version. IBM Touchpen support, which is in the current version as an undocumented feature, will

[119] In retrospect, make that a decade or two.
[120] Good, though.

be officially supported in this release if we get a working Touchpen for IBM by March 1.

AutoCAD-80

An evaluation version of AutoCAD-80 (and AutoCAD-86 on the IBM) is now running on the Vectrix, a beautiful $5,000 670 × 480 display with 511 colors selected out of a menu of 16,000,000. The version for the Aurora is released, and actually sold more copies than the Microangelo version in January.

The program now fills memory pretty thoroughly,[121] and the main development project is to compact it a little more. It is also being fixed to run on multi-user Turbodos systems.

There was some discussion of whether we should fix on PL/I (now that the 8086 version is coming out) or on C as a single language for AutoCAD. C, however, is impossible for AutoCAD-80, while PL/I has much less future than C on the new machines, particularly the 68000. This leaves us stuck with two versions forever.[122]

AutoDesk

Kern Sibbald is turning over the AutoDesk project to Mike Riddle. When Kern returns from vacation, he will start taking over the AutoCAD-80 project. AutoDesk is not close to a release yet, but as noted before, we couldn't effectively push it if it were.

AutoScreen

There was no report, but the subject of conversion to the 8086 was discussed later in the meeting. The problem with conversion, now that CB-86 is supposed to be out, is the 3,000 lines of assembly code that have to be translated. The translation program for converting 8080 assembly code to the upward compatible 8086 is apparently worthless for large programs; therefore, the cost of conversion could be justified only if we had very good prospects for sales.

Corporate

The only thing happening here is an attempt to get the books set up properly. This becomes critical as the date for income tax (75 days after the end of the fiscal year) approaches.

Trade Shows

There was a post-mortem on the two trade shows (CADCON and CPM-83) in January, together with COMDEX.[123]

[121] 64K, that is!

[122] Well, not *forever*. AutoCAD-80 was abandoned in 1984.

[123] Exhibiting at both of these shows was quite an accomplishment. CADCON and CPM-83 were scheduled in the same week. We loaded all the company's computers into a station wagon, and Greg Lutz and I drove to Anaheim and set up the equipment for CADCON

There was a general failure to plan for moving things out on the last day of CPM-83. In the future our policy will be that at closing time of any show we have the Walkermobile[124] and/or the Drakemobile to haul things. Either of these can carry our signs and large chunks of equipment, and is accompanied by someone who is supposed to know what's happening.

There was a poll on the value of the three shows. COMDEX was considered very valuable and successful. CADCON was thoroughly marginal, with a low turnout of not very well informed people; and we didn't properly exploit the chance for a good look at the expensive competition. CPM-83 was much like a Computer Faire; the management of the show was much worse, but there seemed to be a higher concentration of dealers and fewer obnoxious people.

Lack of preparation and discussion before the shows sometimes caused people to feel they were making fools of themselves, as in trying to demonstrate what turned out to be an unimplemented feature. On the whole, we do much better than other exhibitors in giving out accurate information at our booth, but there's still room for improvement. It was suggested that everyone who will be working a show should take the whole of setup day off, so that we can do a tutorial on that day, either before or after setting up. There was no real consensus on whether we have reached the point of diminishing returns, where we're good enough at shows that there's no point in spending more effort that might be useful elsewhere.

We seem to need an AutoCAD Jockey. Like Computervision and other companies, we should find people who are especially good with AutoCAD and make them specialists in making the product look easy and impressive. Jockeys might be recruited from within the company or from end users. Even if they can't go around giving demonstrations, they can create drawings and set up demos for other people. This was agreed to be a good idea, but so far we lack anyone to take on the job.

An especially acute problem at shows is the taking of orders for products. At CPM-83 the mechanics of taking orders were haphazard—it was hard to find a clear space to write on or a place to file things—and we ended up with many undecipherable orders. The ideal solution would be to have one person doing nothing but taking orders; but it seems impossible to do that in a ten by ten booth.

There was agreement on some suggestions: All forms, manuals, VISA slips, etc., will be organized in neat, possibly color-coded boxes. If we have no other tutorial session before the show, there will at least be an indoctrination on the ordering procedures. There should be a special order form with "Take" and "Send" at the top to indicate whether the customer has taken delivery of what he bought. There should be a clipboard or something to guarantee a surface to fill out forms.

There was a time at CPM-83 when there was only one person at the booth. We must keep at least two people in the booth without fail, preferably three at a show full of thieves, like the Computer Faire (four is a crowd). This means that we should have four or five people at the show at any time.

---

at the Disneyland Hotel. Greg, Mike Ford (who flew in late that night), and I shared a room at the Disneyland Hotel. We manned the booth for the show, then tore down everything, loaded it into the station wagon, and drove through the night back to Kensington (near Berkeley), where we dumped all of the equipment in Greg's garage at 4 A.M. The CPM-83 team arrived there shortly thereafter, retrieved the equipment from the garage, and set it up at CPM-83. That show opened the next day, staffed by a team including those who had just awakened after sleeping off CADCON.

[124] We put the ten tons of garbage in the Red V.W. Microbus and....

Manpower

As mentioned before, the Autodesk project is going to Mike Riddle. Kern Sibbald will start taking over AutoCAD-80 when he returns from vacation. Duff Kurland will start doing AutoCAD-86 work and will investigate getting our documentation on a decent word processor, which might be Perfect Writer.[125]

Summary of Decisions

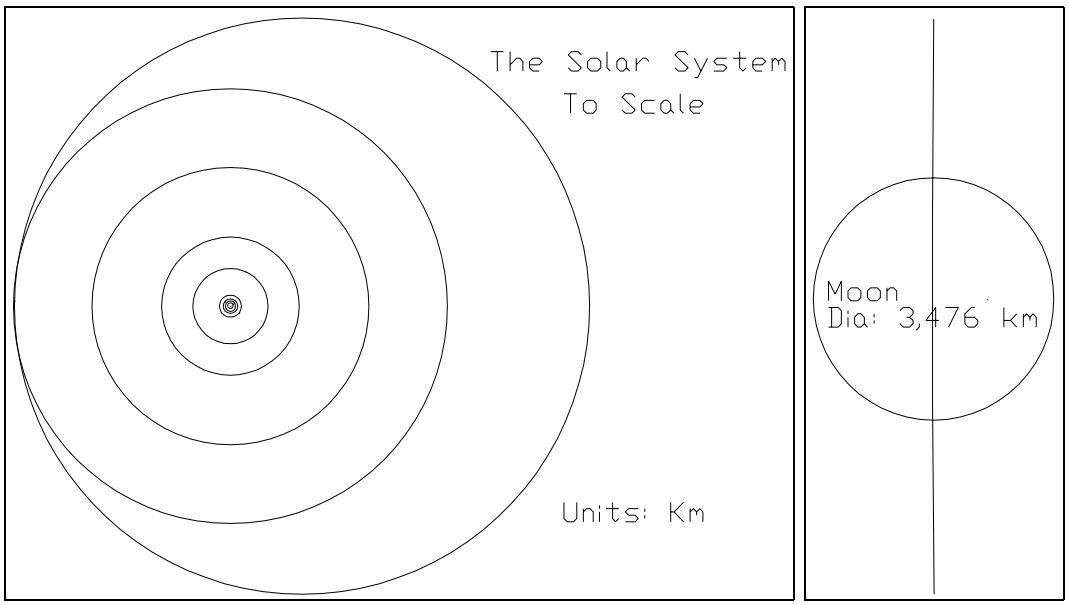
There have been manpower shifts as described above. AutoCAD-86 has a manager (Dan Drake), who should get all phone calls; the people doing technical work are to be left alone.

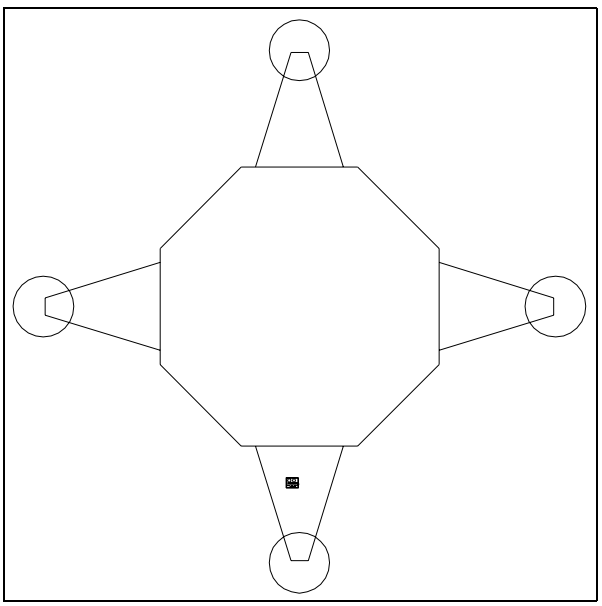
The advertising budget is an ad hoc $1,000 this month, to be regarded as an expenditure of working capital. A real budget will be allocated in a month or two.

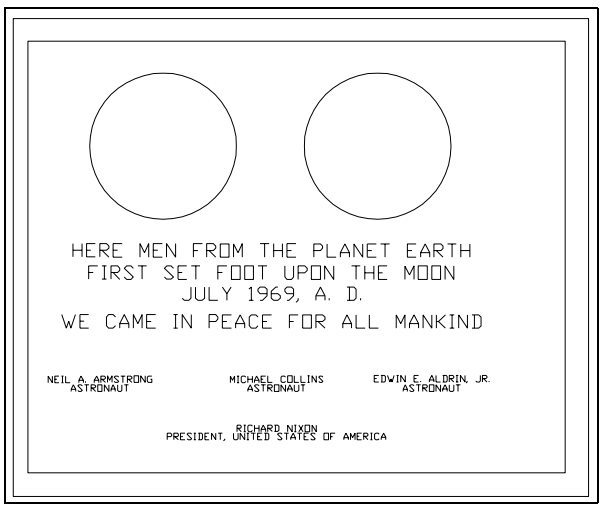
There are specific policies for improving our handling of shows, in terms of coverage of the booth and handling of sales.

[125] As opposed to Marinchip Word, which we had abandoned in favour of the wiles of WordStar. Perfect Writer proved less than its name implied, and was soon thrown over for MicroScript, then LaserScript, then GML/PC, and the saga continues.

The Solar System drawing was our flagship demo of the resolution of our floating point database and formed the centrepiece of many of the demos we did for venture capitalists in 1983. It was also the very first drawing I ever did on AutoCAD-86. I drew it in one afternoon on a Z-100 that I had just gotten to work with AutoCAD-86 (my first work on that version of the product).







March 1983 Meeting

The March, 1983 meeting saw Autodesk rapidly evolving toward an operating company focused on the development and promotion of its hit product, AutoCAD. No regular minutes from the meeting exist—only Dan Drake's notes on what people agreed to do. Those tasks speak eloquently to the direction of the company.

The attached "Software Control Policies" was the first attempt to come to terms with the reality of quality control and effective product management of what was rapidly becoming a mass market product.

Odd Jobs From the March 6 Meeting

by Dan Drake
1983 March 8

We've found several times that people come away from monthly general meetings with different ideas of who has volunteered to do what. We can't afford to discover these disagreements a month later at the next meeting; and the minutes, even if they come out on time, don't cover these matters conspicuously if at all.

This is an attempt to list all the little tasks that were allocated at the last meeting. It doesn't claim to be definitive, but it gives you something to disagree with, preferably to the relevant project manager, if you find yourself listed for the wrong task.

Odd Jobs

1. Odd Job List

Dan Drake will put out this odd job list on the Tuesday afternoon or Wednesday morning after each general meeting, if it seems to be doing any good.

2. Hewlett-Packard Contacts

Duff Kurland and Jack Stuppin will pursue contacts at HP to get us in touch with their plotter people. Dan Drake will do the same through GDL.

3. IGES

Duff Kurland will try to get the Interim Graphics Exchange Standard from NBS.

4. Computer Faire Setup

Dave Kalish, Mauri Laitinen, and Dan Drake will set up the booth on Thursday, March 17.

5. Computer Faire Co-ordination

Roxie Walker will publish a schedule of work assignments for the Faire. She will also set up the mechanisms for selling things and taking orders.

6. Ethernet Links

Dave Kalish will look into any plans that Apple and 3COM may have for supporting the Xerox protocol on Ethernet; we may decide to write such software if bigger people aren't already doing it.

7. Applescreen

Dave Kalish will get a 300 baud modem and get Autoscreen running on the Apple Softcard.

8. Burning the Boats

Everyone in the company who hasn't done it already will send Jack Stuppin a letter giving the conditions under which he's willing to go to work full-time for ADI: the earliest time possible, the minimum subsistence pay, confidence in ADI's continued cash flow, or whatever.

Software Control Policies

Software Control Policies

Dan Drake
Revision 1, 83/3/25

(This is a draft of a draft, on which I'm soliciting suggestions and arguments. In particular, it would be nice to get beta testing done without a six-week delay. Does anyone think I'm too paranoid about beta test?)

Now that ADI is beginning to get substantial sales, we need to face some of the business questions that can trip us up (unwise commitments, continual vacillation on decisions, etc.). Here are some of my ideas on policies that might save us some grief and indecision; they're inspired by thinking about AutoCAD-86, but it seems to me that they apply to any software.

First I'll give some of the motivations, then a list of ideas that I'd like to see agreed on.

Proprietary Rights

AutoCAD is (subject to a percentage royalty) the exclusive property of Autodesk, Inc. Those of us who wear two hats in order to make a living while building the corporation must keep this clearly in mind while wearing the consultant or dealer hat.

Quality Control

We've been determined from the first to uphold the highest standards of quality and reliability. Lacking capital and sufficient marketing staff, we have no other unique edge on the rest of the world.

There are only two or three people in the company with the experience of making a product that goes out into unknown places in the real world (as distinct from software for a service operation). I don't think we have anyone who has sent out a program in hundreds or thousands of copies as we plan to do. In this environment a little bug in a program can mean a little incremental update for vast numbers of customers, which will feel to us like a GM product recall.

Our testing procedures so far have not been rigorous enough. In the future, we need real, bona fide, beta testing. At best, this will have an effect on our product development time that will seem little short of disastrous; we need to make realistic plans that will keep us in the best case.

Proliferation

One of our most important plans is to put the software on every machine that will have any substantial sales, so that manufacturers will in effect have an investment in us and won't look promising to potential competitors. It's important to remember why we're doing this, so that we don't get off track. If a dealer lends us equipment for a conversion, we don't do the job as a friendly gesture, a speculation on selling a couple of dozen more

copies, or even a service for a fee; we do it because a working version of the program may give us a shortcut through the manufacturer's bureaucracy, or at worst produce vast numbers of sales through many dealers.

Meanwhile, every conversion that's released for sale increases our support burden, and every commitment to do a conversion puts our credibility on the line.

Delegation

It's important for everyone to know who makes what decisions. It's not especially important to distinguish between the board of directors and the top management, since they're pretty much the same people; but we need a feel for what the board does and what it lets someone else worry about. I've tried to outline the level at which various decisions should be made.

Proposals

Any technical work that any Autodesk person does on AutoCAD is the property of Autodesk, Inc. The person doing such work is wearing an Autodesk hat and will be compensated by Autodesk's normal procedures.

All decisions on selling the product of such labor (e.g., whether, when, where, and how) belong to the company. Making such decisions is a property right: the "first refusal" provision of the employee agreement does not apply, insofar as the work is based on privileged access to Autodesk's intellectual property.

If the company decides to distribute a piece of work, it will sell to any qualified distributor or dealer who wants it. The only exception might be an exclusive distributorship agreement negotiated at arm's length; such decisions would never be delegated to the low level of a product manager or even a marketing manager.

When policy is to put software on lots of equipment, anyone involved with the product can and should talk to any interested supplier. After the first conversations, the company may decide to make a conversion for evaluation by the manufacturer. No one but the product manager (possibly under orders from above) can make this commitment, and no one should hint otherwise. Our credibility is on the line here.

When we've converted something, the supplier gets an evaluation copy. This will serve as a basis for negotiations on the possible distribution of the product. It must be absolutely clear to the supplier that we are in no way committed ever to release anything to the public. The marketing decision belongs to the company; what this means is that the marketing manager makes the decision with the agreement of the product manager, who will have to provide support.

Obviously, no one will pay us an engineering fee for a conversion without a commitment that we'll let him sell the product. The last paragraph doesn't apply if such a case ever arises.

It is standard policy that we will not support any piece of hardware unless we have a piece of that hardware on indefinite loan. Of course, we will never make a change that introduces a bug into something that worked before, any more than Jimmy Carter would ever lie to you; but we can't prove that a problem isn't our fault unless we can reproduce it first. Besides, as a practical matter, our good repute requires us to be able to reproduce problems that a user reports, and provide a correction, even though we are blameless.

People who are working on new features will be left alone as much as possible, because breaking their concen-

tration squanders the only resource that we have in quantity. Before calling an implementor, consider whether someone else, such as a product manager or a salesman, could do as well. As for the release of a programmer's phone number to anyone outside the company, the general rule is simple: *no*. The handling of exceptions will be up to product managers. In the ACAD-86 game, the product manager is the only person whose phone number can be handed out without his prior explicit consent.

No change in software, however small, can be released for sale or even for beta test without clearance from the product manager. Any manager who does not constitute the entire project staff must work out procedures for source code control and distribute them in writing. (This will raise problems when someone overseas is making versions that change nothing but the language in which the messages are written.)

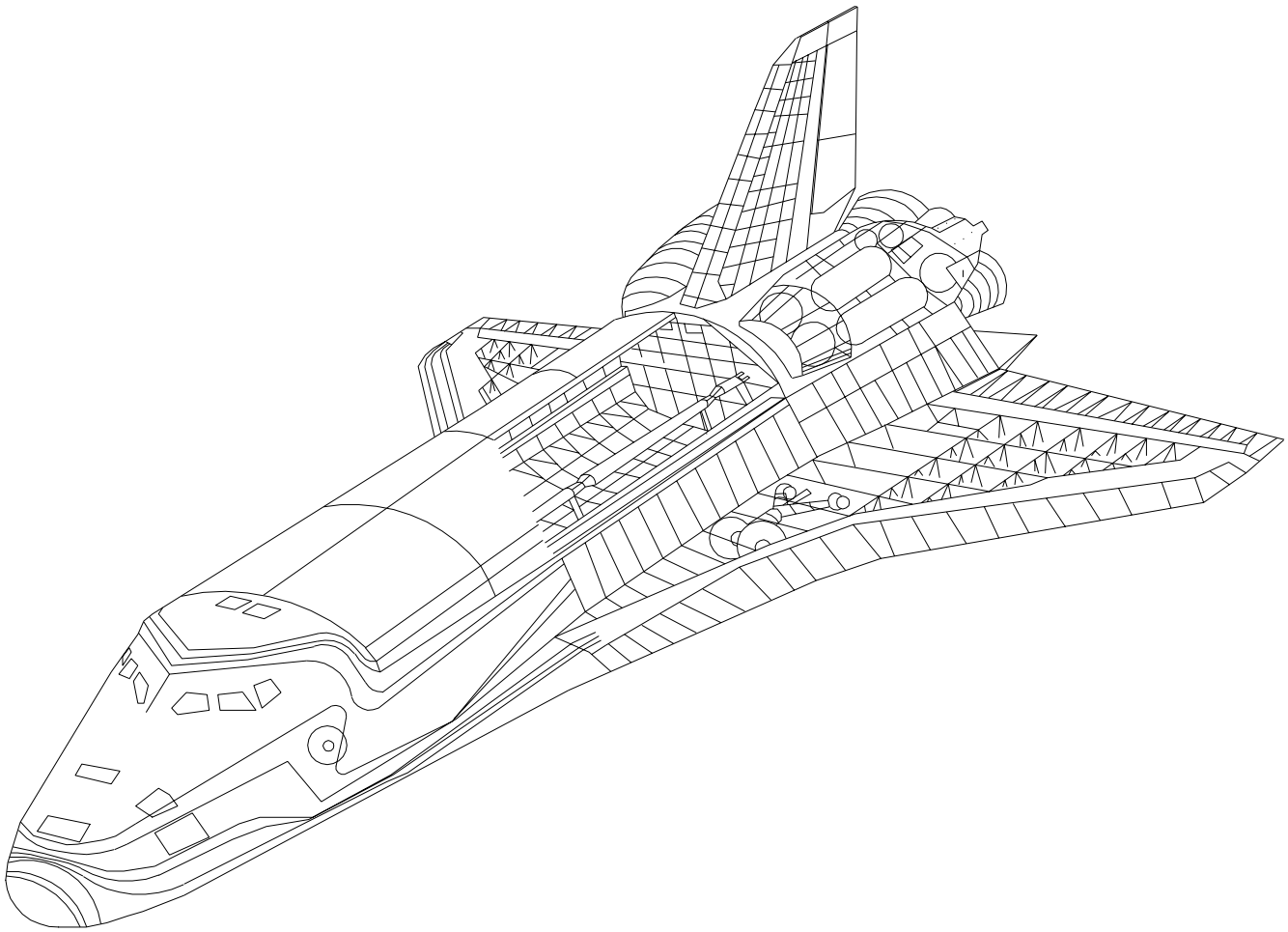
In AutoCAD-86, for example, there will be a clear distinction between a core section and device controllers. One programmer will have primary responsibility for the core code and will do integration and testing of any changes worked out by other people. When a new version is ready, it will be distributed in source and relocatable form to the various people in charge of device drivers, who will integrate their drivers with the supplied relocatable and do thorough testing. If this exposes a need for changes in the core source, the countdown stops while the change is worked out, integrated, and distributed to everyone for the next iteration.

Product managers must also work out plans for beta testing. We need some good ideas here, and we need to live up to them. Fortunately, the iterative integration process in the last paragraph can be overlapped to some extent with beta test.

It appears that the features freeze will be weeks before the release date. It will take a good deal of self discipline to avoid diddling the code as the testing cycle drags on.

The bulk of documentation work should be done, as we all know, at the beginning of a new development, not at the end. Whether we can afford to do this really right is an open question. In any case, the document will not be a last-minute panic project. The documenter will estimate how long the job is to take; the project will then be scheduled so that a draft will be ready when beta test begins. For final proofing there will be a mock camera-ready version two weeks before the release date.

Several people attempted to use early releases of AutoCAD to make this beautiful drawing of the shuttle that graced our sample drawings disc for so long and appears in so many advertisements of display hardware for AutoCAD. This version, the one we finally used, was drawn by Sean O'Donnell.



June 1983 Meeting

The June 1983 meeting was the first general meeting to include all of the Europeans who were actively developing and marketing our products. By that meeting we were clearly profitable and had generated more cash than twice the original investment in the company. Autodesk's stock value was clearly on the march, though nobody suspected how fast and how far it would go.

These notes, taken by Kern Sibbald, also reflect the inevitable growing pains inherent in turning a loose collection of programmers into a real operating company.

Autodesk, Inc.
June Monthly Meeting (June 5, 1983)

by Kern Sibbald
June 7, 1983

Introduction

I have volunteered to take notes of the monthly meetings and to distribute them to everyone with the following ground rules:

1. I'm going to spend near zero time editing and perfecting.
2. I'll write from the notes that I took—no tape recordings.
3. I'll try to complete it within 2–3 days and have it reviewed by at least one other person.
4. I'll mail it out to everyone the second Tuesday following the meeting.

Under these ground rules, please don't expect too much. I am interested in suggestions and feedback.

Attendees

The meeting was held in the cabana at Duff's condominium complex at 1:30. The following people attended: Lars Moureau, Richard Handyside, Rudolf Künzli, Mike Riddle, John Kern, Duff Kurland, Dave Kalish, Mauri Laitinen, Hal Royaltey, Greg Lutz, John Walker, Kern Sibbald, John Nagle, and Dan Drake.

Financial Report: John Walker

Due to the amount of financial activity lately, the report is approximate:

Assets		
Capital Preservation	$115,000	(up 45,000 from last month)
Receivables	42,000	(7 days average age)

Liabilities	
Salary for John Kern	????
Mike Riddle royalty	15,000
Mike Ford commission	15,000

To be more exact, some 370 invoices need to be processed.

Dan gave the following status report on setting up our books: Arthur Young has set up a chart of accounts and almost completed the books for our last fiscal year ending Jan 31. We show a tax loss of about $8000 which we can carry into fiscal 1983.[126] The cost of the accounting service is somewhat unknown (about $1000) because accountants are like lawyers, it is not proper to ask, one simply waits for the bill.[127] Jack Stuppin is, however, negotiating to get us the most for the least. Another 5 months of Autodesk accounting needs to be processed before we are up to date. We will be able to pay our taxes that are due on July 15. The accounting is structured to handle overseas sales.

Board of Directors Report

The cash flow will soon be sufficient to reimburse us for our expenses such as car mileage so *keep good records* if you want to get reimbursed. We are all cautioned about making commitments for the company over the telephone particularly regarding what equipment Autodesk will support. It is OK to indicate that we are working on a machine, however. If you have a machine but don't have everything you need from the manufacturer to make it work, write a note and send it to Mike Ford. He will get you what you need.

[126] If we had known the trick of capitalising our organisational costs, we could have shown a profit for that year, and thus for every year we existed. This is even more remarkable when you remember that in our first year we were only shipping products for a month.

[127] Gentlemen don't discuss money, I guess.

Marketing Report: John Walker filling in for Mike Ford

Several people visited Digital Research in Monterey last week and found that they have machines everywhere and are running around like mad trying to get their software running on all of those machines. The main topic of conversation was apparently GSX, Digital Research's new CP/M graphics device interface. It looks like for AutoCAD, it will provide a universal display driver and a way to integrate mouse drivers, but we will have to continue to write our own plotter drivers. GSX could solve most of our problems supporting so many devices and only costs $60.00. DR is also interested in a $200 drawing package which could possibly be a stripped down version of AutoCAD.

A Sony Z80 system may have been shipped to us.

Valid Logic Systems is interested in AutoCAD. They have a very high resolution graphics device on a Unix based 68000. Keith talked to them and they agreed to send him a machine and pay a $2500 engineering fee. We hope to get TROFF (Typesetting Runoff) with it.

NCR has a machine on the way.

USI still wants 1000 copies of AutoCAD for evaluation and marketing (maybe?). However, they have not yet shipped any OptoMouses because of a bug in the production ROM. They seem to be finding their way out through cracks.

We are negotiating with a number of schools. U. of Idaho will buy one ACAD and get 9 more for developing a tutorial. U. of Arizona has 30 PC's and wants a site license; we are investigating.

Mike Ford has prepared a dealer contract that was circulated for comments at the meeting.

We have rewritten the end user license, which will be one sixth the size. It has been lifted from Victor; is very nice.

The registration card is being modified to have the machine serial number on it and possibly the new license since it is so short.

We are pursuing a hardware lock for AutoCAD—seems like pure blue sky at this point.[128]

John is exploring a deal involving mapping archæology that may lead to a write up of AutoCAD in *LIFE* magazine.[129]

Chuck Victory is putting together a 7×10 ad for *InfoWorld*. We are in the September issue of *PC World* in the directory under graphics.

We put on a very well received demo for the AIA (American Institute of Architects) in San Francisco.

We traded an AutoCAD to Victor for an 8087.

[128] This was a gizmo where the program printed a number which you keyed into a little calculator-style gadget. It displayed a password, which you then typed into the program. Dumb idea; we didn't pursue it further.

[129] It was this project that caused John Walker to waste endless hours on Otrona versions of AutoCAD-80 *and* AutoCAD-86.

Reports from Europeans

Richard

He has given over 110 demos. AutoCAD has been well received at many shows. Approximately 60 dealers have been found; 50 of them have been to Richard's house. So far sales amount to 38–36 Sirius, 1 IBM, 1 Z80. Richard spoke at length about the different "real" users with quite a few well known names such as Ford Motor, Shell, British Royal, etc. He has had one and a half people helping him since January to answer the phone, etc. He is spending a lot of time on support problems. To keep our happy customers happy, he feels we must solve the REPEAT problems and develop several required new features. There is even one software house with their own CAD package that is now selling AutoCAD. A firm with 200 draftsmen is organizing a course to teach computer drafting with AutoCAD.

Lars

There is high interest in AutoCAD but few orders for several reasons. The users want the source code safeguarded in Sweden (against war, etc.); they will wait to see if it continues to be popular; they want a Swedish language version. Norway and Denmark will follow Sweden in sales. He has ordered 13 systems and 5–10 demos and projects 100 copies by the end of the year. He says the first year will be the hardest until it is an established product. One company alone is talking about 50 Z80 copies. AutoScreen is hard to sell because WordStar took the market.

Rudolf

The French language manual will be ready in 2 weeks, the Italian version is in translation and should be finished in about 1 month.

DEC Rainbow promises to be a big seller in Europe because of its support.

We have had a number of very good press releases including one (not yet published?) by a German CAD magazine with a circulation of 200,000.

There were some good war stories about transporting equipment across borders for the Hannover Fair which was attended by Richard, Lars, and Rudolf and had 500,000 people in 10 days.

Domestic Sales Summary

400 approx. shipped. 370 to SunFlex as of June 2.[130]

[130]Since the SunFlex versions were all Victor, Victor sales accounted for more than 90% of our business to date.

Corporate Status Report

Lars' option has been signed and the certificate will be issued soon. We are trying to set up a European company wholly owned by the U.S. company. Jack is heading up this effort.

Operational Report: John Kern

Our office is now functional and is located at 150 Shoreline, Bldg B, Room 20. The phone number is (415) 331–0356.[131] We are looking for a larger one so we can do demos there. 95–97% of the calls are handled by John Kern; the others go to Dan or Duff. All Victor disks are made there. IBMs are made at Mike's house for now but soon we will have another IBM and make disks in the office. We have a new perfect-bound Victor manual—very slick. John Kern turned around 3200 bingo cards in the first 3 weeks. We need complicated good looking professional drawings done on AutoCAD to help sales to architects. Office Staffing—John Kern full time, Kathy Marcelius half time, Jane Kern part time, Gladys Sibbald part time.

Customer Support

We will talk to Kevin[132] soon. Jack Stuppin has put us in touch with a woman who is a prospective customer support rep. We have a job opening for a customer support representative for $1500/month. If anybody is interested now is the time to apply.

Incentive Options

The plan submitted was approved 2 days after the May meeting.

AutoCAD-86 Status Report

The conversions are progressing nicely. John Walker and Kern Sibbald are working on the CompuPro. The DMS 5000 will show in Texas on the 6th. Greg is working on a configurator, which was called a "crash project to keep us from going under".[133] He is approaching the beta test stage. A lot was said here about the configurator. We need to get the Victor version running on MS-DOS[134]. Mauri has the Eagle technical documentation for the Eagle now and hopes to get something going. Duff is working on the NEC; he has the display driver part working and gave us a demo after the meeting. Dave has been working on the TI and

[131] Amazingly, more than ten years later, in late 1993, this number still works.

[132] O'Lone

[133] Until this project was completed, a different AutoCAD had to be built for every combination of display, pointing device, and plotter. Really!

[134] It was initially done under CP/M-86.

has completed the code and must now integrate it into AutoCAD. He is looking for a debugger. John Walker finished the Z100 3 hours before Mike left for NCC[135]. Richard has a working HP[136] driver.

AutoCAD-80

John finished the HP[137] driver. He has shipped a bunch of ACAD-80's and sold 10 last month. He would like to drop support of the Z80 version but finds it difficult with several stories of orders of 500 possible.

The Digital Research C looks like a winner.[138] We will get it late this month.[139]

AutoCAD-68K

No floating point now or likely to come in the C we have from Digital Research. However, there is a full C due in the fall from another vendor.[140]

Prometrix

For $15K we can buy 2% of this company which reportedly has 7 orders for a machine that finds defects in chips??? (I was in the head when this was discussed). John is interested to get input about this venture. It won't happen immediately.[141]

Company Problems

The original company is dead. We are now a classic small company with one product rather than the imagined company developing many products. In any case, we must now act like a real company to succeed. We discussed the fact that it would not be possible to run a real company by getting everyone's consensus. After a lot of discussion, it seemed like everybody agreed that for operational decisions such as hiring and firing and getting facilities, etc. that the decisions must be made by the board of directors and the officers as they would in a normal company. However, strategic decisions such as bringing someone new into the company as an owner, or going public with the stock or other such major decisions should be discussed by everyone. The biggest complaint seemed to be the lack of communications about decisions rather than the fact that the decisions were made or how decisions were made. Hopefully notes like these and more participation in the company as it prospers will help.

[135] This was his first tilt with MS-DOS.
[136] Plotter. For many more months, we supported only Houston Instrument plotters.
[137] Plotter.
[138] It wasn't.
[139] We didn't.
[140] This referred to a CP/M-68K version to run on the CompuPro 68000 CPU. Nothing ever came of this version.
[141] It never did. They made a device to characterise silicon wafers and find defects. This is how we met Dick Elkis.

We need to formalize the bug reporting process. Dave Kalish volunteered to keep an error log and to give it to Kern monthly for distribution.

SunFlex

We meet with SunFlex Thursday to resolve the European question. One point seems to be to insure that we are adequately compensated for the extensive marketing we have done to date.

Autodesk: Mike Riddle

How can we sell it? It is not very fast. No overall architecture. It still has one major bug. He doesn't recommend selling it. It is dead.

Autoterm: Rudolf

Stopped completely.

Resumes

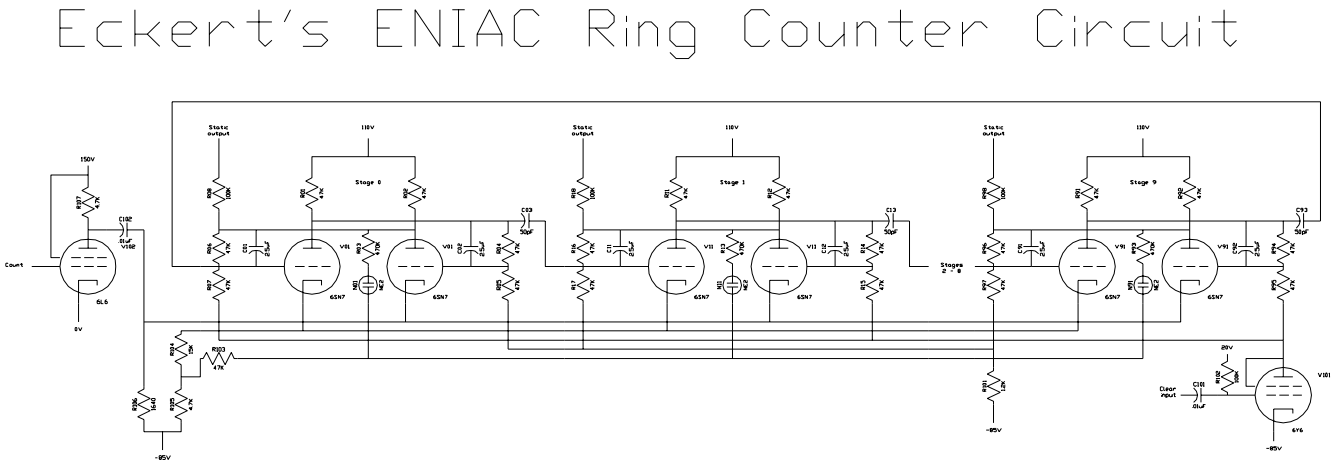
If you haven't done so, send your resumes to Mike Ford. Mike's sales plan is almost spot on so he is going to test his forecasting record by writing up a business plan.

Equipment List

Please send me a current list of the equipment you have and the projects you are working on and I'll collect it all and publish it in the next notes.

Kern

I made this drawing of the ENIAC ring counter circuit to test the new Attribute facility prior to the release of AutoCAD 2.0 and to serve as the testbed for the sample attribute processing BASIC programs I was writing for inclusion in the AutoCAD manual. This was the first drawing I ever recall doing on an IBM PC.



AutoCAD Wish List

We'd always hoped that once we got a product into the market, our customers would direct our development efforts through their requests for enhancements to the product. We couldn't have wished for a more energetic, imaginative, or vocal community of users. After six months of shipping AutoCAD, the lists of requests for new features were growing so long that we decided to get together, merge all of our private lists, and try to sort them by size of the job and importance of the feature. Duff Kurland prepared this first-ever AutoCAD Wish List from his notes of this meeting. Duff continued to be the keeper of the wish list for several years thereafter.

Any doubts about the veracity of our claim "our development agenda is taken directly from the list of user-requested features" can easily be dispelled by comparing with wish list with the features in AutoCAD releases up to the present day. I've added annotations in italics listing the release in which each item was eventually implemented.

AutoCAD Wish List

by Duff Kurland
Revision 0 — June 10, 1983

Introduction

An AutoCAD enhancement technical session was held on Monday, June 6, 1983, at John Walker's home. Present were Dan Drake, Richard Handyside, Rudolf Künzli, Duff Kurland, Greg Lutz, Lars Moureau, Mike Riddle, and John Walker.

A list of desirable features was compiled and discussed with varying degrees of detail. An attempt was made to prioritize these items, and some were assigned to individuals for implementation. This document has been prepared so that those who were not at the meeting (and those who were) will have a basic understanding of what's going on, and what the project names mean.

A few general notes are in order before presenting the list of features. First of all, AutoCAD-80 is *not* expected to be enhanced at all. Secondly, the priorities were set based on a combination of factors:

- Are we losing sales because we don't have this feature?

- Is this a "snazzy" gimmick feature which could attract additional sales?
- Does this feature reduce user confusion and our support burden?
- Could this feature be easily implemented?
- Would implementation of this feature make it easier to implement some of the other features?

Lastly, strict priorities have not been set. Some of the low priority items may actually be among the first done, if they're in areas where we're already poking around.

I have included notes on the discussion of most items, but they are by no means complete. I would welcome comments, clarifications, and additions; this list will be continually updated, and published at reasonable intervals.

High priority "quick kills"

Alternate arc specification

The ability to draw an arc by specifying its center, radius, and start/end angles has been requested by users. This is somewhat embarrassing; that's the way we encode arcs internally, but the user cannot specify them that way. Other combinations, such as endpoints and included angle have been requested, also. *1.4*.

Text size by length

This is the ability to select the text size based on the length of the field in which it is to fit. *1.4*.

Layer-to-layer move

AutoCAD-80 now has a "**`CHANGE LAYER`**" command to allow selected entities' assigned layers to be changed. A similar capability is needed in AutoCAD-86. *1.3*.

Standard drawing config setup

This item was discussed briefly, and I'm not sure what it encompasses. Discussion included the ability to select the size and resolution of a new drawing without prompting the user for the details each time. Two methods were proposed; selecting defaults via the new "Configure AutoCAD" main menu item, or allowing the user to specify the size using ANSI or DIN sizes with a default resolution. A more elaborate "drawing type" scheme was also proposed (see "Questionable items" below). *Prototype drawings in 2.1*.

Drawing header to DXF file

Drawing interchange files do not currently contain certain information about the drawing (insertion base point, etc.). This information is in the drawing file header, and should be added to the DXF file. *1.3*.

XOR grids when possible

This would be a change to the display drivers ("**dsdot**") to invert the pixel at each grid point, rather than simply set it. The idea is to ensure that the grid is visible even on a filled solid area. (Note: "**GRID ON**" will currently write a grid even if the grid is already on. This will have to be fixed first.) *1.3*.

Change "REDRAW ON/OFF" to "FILL ON/OFF"

This will avoid two areas of confusion, since "**FILL**" is a better description of the command's effect, and "**REDRAW**" won't perform different tasks depending on which key (space/return) is used to terminate it. *1.3*.

Change "RES" to "SNAP"

This should also eliminate some user confusion. *1.3*.

Change "P1, P2" point prompts

The "**SOLID**" command should prompt with "**1st point:**", "**2nd point:**", etc. as documented. *1.3*.

Change "Cmd:" to "Command:"

User friendliness (eschew obfuscation). *1.3*.

Enhanced HELP facility

I forgot to bring this up at the meeting, but feel it belongs in this category. First, "**HELP**" should be a synonym for "**?**". Second, we should support requests such as "**HELP CIRCLE**", which would display information about the **CIRCLE** command. I've already written an extended **HELP** file to support this capability. *1.3*.

INSERT angle governed by ORTHO

If **ORTHO** mode is on when an object is **INSERT**ed, the insertion angle should be constrained to 0, 90, 180, or 270 degrees. *1.4*.

Stop using square brackets

Several AutoCAD prompts display the current value within square brackets. Unfortunately, these character codes are used for foreign language letters. We will change to angle brackets. *1.3*.

High priority larger items

Polylines

A polyline is a group of lines, gaps, and arcs (?) which are associated with one another. They can be edited to add, delete, or move a vertex, move a line segment, etc. A width should be associated with the polyline; perhaps double walls could be special polylines. Assigned to Duff Kurland. *Done by Dan Drake in 2.1*.

Cross-hatch/pattern-fill

John Walker has been experimenting with a cross-hatching technique which seems to work. We should implement the standard hatching patterns for various structural materials (concrete, steel, mud, etc.), and should consider a general user-defined pattern fill capability. Would be an extra-cost option. The project has been assigned to Mike Riddle. *Done by John Walker in 1.4*.

Splines

John Walker has also been researching various spline drawing methods. We had hoped that IGES would point us in the right direction here, but it doesn't point anywhere. *Release 9*.

Double walls

Architects require this feature. A center line capability is also needed. Polylines might do the job here. *Provided in AEC*.

Line types & color

Several topics are covered by this item. First, we need to standardize on our color representations. For instance, the first eight colors should be:

0 black (erase)
1 red
2 green
3 blue
4 cyan
5 yellow
6 magenta
7 white (black on plotter?)

On monochrome devices, 0 means black (off), and any nonzero value means white (on).

Up until now, some AutoCAD implementations have used various bits of the "color" number to select the dotted/dashed line features of hardware devices (Scion Microangelo, NEC APC, plotters). While this has the desirable effect of allowing monochrome displays to differentiate between colors, it has two undesirable effects and must be avoided. First, it tends to make the color numbers difficult to work with (red + dashed line = 1 + 32 = 33). Second, it conflicts with the need for standardized line types.

One area which was not discussed at the meeting was the choice of colors for things AutoCAD (not the user) draws, like crosshairs and grids. My feeling is that the crosshairs should always be white, while the grid might be best in green. *1.3*.

Geometric snap

This is the ability to draw a line which intersects another entity in some specified manner (tangent to arc, perpendicular to line, etc.). *2.0*.

Breaking walls/partial delete

It should be possible to select two points on a line, and split the line into two segments with a gap spanning the two selected points. This should not be limited to simple lines, however. Polylines, walls, traces, circles, and arcs should be breakable. *2.0. Polylines: 2.5*.

Fillets

Fillets are arcs which smoothly connect two lines. We should have a method of applying fillets after the lines have been drawn, and a method (FLINE command, or POLY command) of drawing them on the fly. *1.4*.

IGES support

Creation and reading of IGES-format interchange files should be implemented. Could be an extra-cost option. Seen as large design project with quick implementation involving adaptation of **`DIFIN`** and **`DIFOUT`** functions.[142] Assigned to Peter Goldmann. *Done by Ben Halpern and John Walker in 2.5*.

[142] Later renamed **`DXFIN`** and **`DXFOUT`**.

Block output

Currently, our **BLOCK** command allows dynamic creation of a new block, but the new block is **INSERT**able only in the current drawing. We need a way to write the block to a new drawing file, so that it may be **INSERT**ed in other drawings as well. *1.4*.

Redefining blocks

Once a block has been **INSERT**ed in a drawing or created via a **BLOCK** command, its definition rides around in the drawing file. In one respect, this is nice; the drawing file for the **INSERT**ed part need not be present after the initial **INSERT** is done. However, it makes it difficult to update the part definition in all the drawings which include it. Even if all references to the block are erased from the drawing, the definition remains; the only way to delete it is to write a DXF file, edit it to remove the block definition, and load the DXF file back in. This is awkward. We need a way to delete or redefine an existing block definition. *1.4*.

Complete dimensioning

Our dimensioning facility can only draw horizontal and vertical dimensions. Several additional capabilities have been requested:

- Angular dimensions
- Arc length
- Circle/arc radius
- Circle center lines

2.0.

Large plotters (32K problem)

Our internal coordinate system uses 16-bit integers, giving a range of 0–32767 points in the X and Y directions. We are now seeing large (48-inch) plotters with 0.001 inch resolution. We need to support them, but they exceed our limits. A workaround might be to use only half of the plotter's resolution for the time being. *1.3*.

Generic user manual

So far, we've been producing a custom user manual for each machine implementation. This probably cannot continue. The basic reasons for separate manuals up to this point have been:

- Differing operating systems.

- Differing cursor control and function keys.
- Commands (**QPLOT**, **PALETTE**), which operate only on some machines.

It might be best for us to produce a generic AutoCAD-86 manual, documenting all the commands, and control keys which will work on *every* machine. I would suggest the following keys:

Cursor left	CTRL-H
Cursor right	CTRL-L
Cursor up	CTRL-K
Cursor down	CTRL-J
Flip screens	ESC 1
Select graphic cursor	ESC 2
Select menu cursor	ESC 3
Return to keyboard	ESC 4
Slow cursor	ESC 5
Fast cursor	ESC 6

A note such as "on some machines, the CTRL key is marked ALT; see the AutoCAD installation/user guide for your machine" could be added. Operating system differences would be noted, as well. A separate installation/user guide and reference card would be associated with each machine, and would include exceptions from the main user manual and a list of alternate function keys if applicable. *1.4*.

Function keys on reference card

The AutoCAD reference card for each machine should include a list of the function keys available on that machine.

Foreign language versions

Rudolf Künzli has been working on various foreign language versions of AutoCAD, translating not only the user manual, but also the messages generated by the program. As things stand, he must re-apply his changes each time we send him new source disks.

We decided to use compile-time tests for each language, so that the text of each message could be provided once and maintained in the master source files. *1.3, later redone using the automatic translation utility*.

Lower priority items

Point variables

This is the ability to attach a name to a designated point, and to use that name in subsequent relative coordinate specifications, geometric snaps, etc. *2.1, via Variables and Expressions, later AutoLisp*.

Extended entity selection

This is the ability to more finely describe the entities to be selected. Possible additional criteria would be layer, color, entity tag (see below), and entity type. Mike Riddle has already done some work in this area. *Done by Kern Sibbald in Release 9.*

Entity tags

These are text items which would be carried around with each entity drawn. They could be used to construct a bill of materials. *2.0 Attributes*.

"Toy" bill of materials

A sample program was suggested to demonstrate the capabilities of DXF files (or was it for entity tags?). *2.0.*

EDIT command

This would be an extended **CHANGE** command, to allow modification of any of the properties of an existing entity. *Extension of the* **CHANGE** *command from 1.3 to Release 9.*

Extended **OOPS** (**UNDO**)

The **OOPS** command restores the last thing(s) which were erased. We need a more general ability to "undo" the previous command (e.g., **MOVE**). *2.5.*

Rejecting added entities

In some systems, the user can try drawing an entity; if it doesn't turn out as desired, he can reject it and try again. For continue commands like **LINE**, this seems like a nice approach. *2.0 for lines, 2.5* **UNDO** *for everything else*.

Repeat last selection

Currently, the "**L**" modifier allows the user to select the last entity in the redraw file. A more general ability to select the same set of entities as most recently selected would be useful. *2.5.*

New **LAYER** command

The current **LAYER** command, with its embedded **COLOR** option, is confusing to users and should be reworked. *Ongoing process. Dialogue box introduced in Release 9.*

GRID enhancements

Our current **GRID** command produces a square grid with specified spacing (within certain limits), with the grid origin at (0,0). We have been asked to provide grids with differing X and Y spacing, isometric grids, offset and rotation capabilities, and something better than the "5 to 50" dot limits. *2.0.*

SNAP enhancements

Similar to the above **GRID** enhancements. Differing X and Y spacing, isometric snaps (or is that isometric **ORTHO**?), offset, rotation. Also, the ability to snap to the nearest of a list of arbitrary points. *2.0.*

Parts library

Some systems can display not only a list of the available drawing parts, but a sample of each one. This is desirable. *Release 9.*

File system interface

To list a disk directory or delete a file, it is first necessary to exit AutoCAD. These facilities should be provided while in the Drawing Editor. *1.4.*

Global coordinate transform

This would allow the user to rotate the display to work on a section of his drawing which is not easily visualized horizontally. *Release 10.*

ELLIPSE command

Currently, the only way to draw an ellipse is to create a **CIRCLE** block and **INSERT** it with adjusted X and Y scales. *2.5.*

Direct commands vs. INSERT

Anything which can be done via **INSERT** should be possible via ordinary commands (see **ELLIPSE**, above).

Transformations and **INSERT ***

Allow scale factors and rotation to be applied to the individual entities in an "**INSERT ***". *2.5*.

Right-justified text

We can now left-justify and center text fields. Right-justification would complete the set. *1.3*.

Feet & inches

Architects like to work with feet and inches. We should be able to handle them in input, and display them in **STATUS**, **LIMITS**, **DIST**, and **DIM** command outputs. *1.4*.

Names for internal variables

Names should be assigned to many of AutoCAD's internal variables, and commands implemented to display and change their values. Some of the names could be documented for users, while others would remain secret for development and debugging. *2.1*.

Menu/keyboard macros

Discussion here included "smart parts" and "parametric entities", which would prompt the user for any needed parameters and use those parameters in expressions. It was also felt that a good macro feature would enable us to create all sorts of new entities easily. Perhaps more importantly, the users could create them also, taking some of the burden off us. *AutoLisp in 2.1*.

Redefine machine interface

Now that we've done a few conversions and have the package running on a variety of machines, we should take a careful look at the device driver routines, with an eye toward restructuring them. Some new common service routines might reduce the work needed for future conversions. *Ongoing process*.

Mode status display

Users sometimes forget what layer they're on, and whether or not **SNAP** or **ORTHO** is in effect. Use of the bottom right-hand corner of the display to indicate mode settings was proposed. *1.3, improved in 1.4*.

Asynchronous mode switches

When drawing something like a continued sequence of lines, it is sometimes necessary to **SNAP** or **ORTHO** only some of the segments. Currently, the user must end the **LINE** command, issue the appropriate mode command, and begin a new **LINE** command. We could provide control keys to allow mode switching during a command. *1.4.*

Arc traces—doughnuts

Again, this might fall under the general polyline-with-width implementation. *2.1. Doughnut command in 2.5.*

Various cross-hair types

Some hardware displays can draw "rubber band" lines and rectangles very quickly. A rubber band could be used along with the cross-hair when entering the "to" point of a line or trace, and when pointing to indicate a rotation angle. A rectangle could be used when selecting the objects in a window. The core program could indicate the preferred cross-hair type, and the base point, to the "**DSMARK**" routine, which would draw a normal cross-hair if it couldn't do the preferred type. "**DSMARK**" would save the necessary information so that "**DSCMRK**" could clear the previous cross-hair when needed. *1.3.*

Enhanced text fonts

An ability to add a slant of a specified angle to an existing text font would be useful, but we should avoid prompting the user for too many things; the **TEXT** command already asks for insertion point, height, and angle as well as for the text string. *2.0.*

Some design work has been done on a new capability for text font definitions (to support more than just 16 vector directions), and some fancier text fonts, including italic, have been constructed and are waiting for this feature. *1.4.*

Multiple text fonts

The **LOAD** command permits the user to load a new text font at any time. What we don't tell him in the manual is that the next time he **REGEN**s the drawing, all his old text will now appear in the new font. Only one font at a time is actually supported. We should look into adding a multi-font capability. Again, we should be careful not to overload the user with prompts from the **TEXT** command. *2.0.*

ZOOM/LIMITS confusion

Our numeric **ZOOM** factors are confusing to users. "**ZOOM 2**" does not necessarily mean "double the size"; it is relative to the original drawing size, not the current display.

Also, "**ZOOM 0.1**" might result in a small drawing in the lower left corner of the screen, and a subsequent "**ZOOM 1**" might leave you with a blank screen. *1.4.*

Views

It was proposed that the user could assign "view" numbers to various portions of his drawing (with associated zoom, etc.). This would allow switching from one area to another rapidly, without the need for several **PAN** or **ZOOM** commands. This might fit in nicely with the "point variable" feature (e.g., "**VIEW KITCHEN**"). *2.0.*

Don't regen invisible layers

Performance optimization. *Freeze and thaw in 2.1.*

Rework REPEAT

The **REPEAT/ENDREP** facility is limited, and can cause confusing results. A capability to form a radial array would be useful. *Array command in 1.4* **REPEAT/ENDREP** *removed in 2.5.*

Generalize redraw files

Currently, our redraw file contains only vectors. Circles, for example, are composed of many small vectors, and cannot utilize the circle-drawing capabilities of various displays and plotters. Even if we could use these hardware features, we'd still need a way to identify such an object when it is pointed to; this currently depends on the vector approach.

Area and perimeter

This is the ability to simply select a polygon (polyline) and compute its area or perimeter. Our present **AREA** command requires the user to specify the polygon vertex by vertex. *2.6.*

QPLOT for additional printers

Currently, **QPLOT** operates only on Epson printers. Other dot matrix printers are popular as well, and could conceivably be used. This might require additional code in the new Configurator. *General printer plotter support added in 2.1.*

3D

A three-dimensional capability is desirable. It appears that an "extrusion" feature might be relatively simple to implement and sufficient for some users. Could be an extra-cost option. *2.1*.

Questionable items

These are items whose value to the program is questionable, or for which additional research is needed before we decide to implement them.

Should entities have colors?

Should color be associated with an entity rather than with a layer? *2.5*.

Aligned dimensions

Although the ANSI standard specifies that unidirectional dimension text is preferable, we have been asked for the ability to have the dimension text aligned with the dimension lines. *1.4*.

Ex post facto `SNAP`

This would allow the user to "sketch" his drawing just as he would on paper, without regard to precision. Once the sketch is done, it could be **`SNAP`**ped (or even **`ORTHO`**ed) into a precise drawing.

Display snapped crosshair

When **`SNAP`** mode is on, some systems only move the crosshairs from one snap point to the next. This makes it very evident that **`SNAP`** mode is on. *1.4*.

Relational entities

???

Display axes

??? *We might not have known what it was, but that didn't stop us from putting it in 1.4*.

Drawing types

A general "drawing type" facility was proposed. A drawing type could have an associated default drawing size, resolution, menu file, and even a skeleton drawing (such as ANSI title boxes). *2.1*.

Alignment of entities

Some systems allow you to draw several boxes, for example, and then adjust them so that their top lines align horizontally.

Shape dragging

This is the ability to move an object across the screen with the cross-hairs in real time, as opposed to erasing it and redrawing it in its new location, as we do now. "If it can be done on an Apple, we should be able to do it on our machines." *2.0*.

DXF to CalComp program

This wasn't discussed at the meeting, but I've had a couple of user requests for it. These guys have large mainframe systems with large CalComp plotters, and don't want to buy another plotter to hook up to their AutoCAD system. We tell them about DXF files, and they ask if we have a program (or know of one) to do the job. The CalComp subroutine package is used widely enough that it might make sense for us to provide a "sample" FORTRAN program, but we'd have to supply the source, and support could become a problem.

Crisis Letter

Everybody involved with Autodesk from its inception has their own list of mental milestones passed on the road from the organisation meeting to where the company is today. The introduction of AutoCAD at COMDEX in 1982 is a key point in everybody's mind. To me, this letter marks an inflection point in the company's trajectory which is just as significant. I believe it marks the transition from an amorphous group of programmers working on many different ideas to a serious, professional company composed of full-time people, dedicated to making its star product, AutoCAD, the technological and sales leader, and making the most of the tremendous opportunity created by its enormous initial success.

This letter marks the time when the change began, but the letter did not cause the change. The transformation of the company was wrought by numerous people, working long, largely uncompensated hours, at daunting and unfamiliar tasks, as the workload only piled higher. Progress could be measured only by realising in those precious few moments available for reflection that the workload which was now crushing us to the point of collapse had, in fact, tripled compared to the workload crushing us three months before. So things had to be getting better.

And they were—and still are. Almost two years to the day after this letter was written, Autodesk completed its initial public stock offering; after that offering, the market valuation of the company was in excess of 70 million dollars. Three years after this letter, Autodesk was named the "Number One Hot Growth Company in America" by Business Week*, and in the very next year Autodesk became the first company ever to win that award twice. The day this letter was written, the value of the company was about* $*200,000. Precisely four years later, the company's market value was in excess of* $*500,000,000. By 1991, its value exceeded* $*1.4 billion.*

Autodesk, Inc.
150 Shoreline Highway, B/20
Mill Valley, CA 94941

June 21, 1983

Dear Autodesk Shareholder,

I am writing you this letter because I feel that our company is in a very deep crisis, and I want to share with you my feeling of urgency about the problems we face and the actions we must take to resolve them. I hope to

communicate to you how critical it is that you act immediately to help save the company and your investment in it.

Today, Autodesk faces probably the deepest crisis in its short existence. We have encountered already many of the problems that all small companies face in the process of growth. We have solved them all, since people involved in the day to day operation of the company were able to focus their efforts on each problem as it arose and track it until it was resolved.

Today, Autodesk resembles the legendary one hoss shay more than a car with one flat tire. Because of our success in marketing AutoCAD, and in stirring up interest in dealers and OEMs, and gaining publicity and reviews of the software, and in closing the marketing deal with Sun-Flex, each single segment of the company is overloaded to the point of collapse. Our technical department has produced and delivered to the field only two features which were on the wish list as of November 1982 (dimensioning and plotter configuration). We have released *no* major new host machine versions since the introduction of the package (although we have introduced additional displays for the IBM and Z-80 versions).

Our marketing department is overloaded to the point that we cannot return all the calls from people who desperately want to do business with us, no less plan a coherent advertising campaign or exert the effort required to make a show work best for us. Our brochures and promotional material are among the most amateurish in the business.

Our front office and production department are overloaded so badly that customers are giving up on Autodesk after trying to get through on the phone for periods of 4 hours. We cannot pack the orders while the phone is ringing, and cannot take orders if it is off the hook. We have not had the time to establish a coherent inventory control system so that we do not run out of critical materials needed to fill orders.

We do not have a customer support group. The service we have given our customers has been very spotty. The service we have given Sun-Flex, our largest base load customer, has been very poor at times.

We are still a long way away from getting caught up on our accounting for the business to date, and longer still from implementing a true automatic accounting system. Thus we still have no idea how much money we have made on our sales to date, and no real idea whether we are making or losing money selling, say, a demo disc.

We have no business plan, even an informal one. We have no budgets for departments, and no way to coherently authorise expenditures or to hire people. The management has not been given a mandate to hire people with stock options, or to in other ways commit the company's assets without fear of recriminations and lost time in argument.

This is a prescription for disaster. This company may be out of business within 60 days.

While we have been sitting dead in the water, others have been introducing their CAD packages. The article in *PC World* which discusses AutoCAD also describes "The Drawing Processor", which compares very favorably with AutoCAD, and has several wish list features AutoCAD lacks. At the PC Faire last weekend, a new product "P-CAD" was introduced. Running on an IBM PC in dual screen mode, it runs quite a bit faster than ACAD, has a very nice menu and submenu feature, and in general looks like a strong competitor. It is priced at $1200. Perusing the 6 page, 4 color, illustrated brochure, one is struck by the appearance on the cover of a Sun-Flex Touch Pen and "CAD-PAD", which is a product under development which Sun-Flex *has not even given us for evaluation*. Clearly Sun-Flex is in touch with these people and negotiating for a better deal than with Autodesk.

And Sun-Flex is the source of the stream of revenue which pays our salaries, allows us to expand our operation, and make commitments to people to come to work for this company on a full time basis. *If we lose it at this point, we have had it.*

So what do we do?

Well, we can give up. In fact, viewed from the outside, there is the strong perception that that is what we've already done. But I'd like to consider the alternative.

This is the fulcrum, the crisis point, the first "crunch". Your actions, and our collective actions, taken now, have an enormously magnified effect 6 months, 2 years, or 10 years in the future. If we go on as a business as usual operation, or as passive shareholders, we can be guaranteed that the number of stock options we own will not be a source of worry in the future—an option to buy stock at $1 loses a lot of its gloss when the underlying stock is selling for $0.

This is the time to neglect your job.[143] This is the time to take that leave of absence from the foundry and work for Autodesk. Spend that long awaited vacation in front of the terminal. This is the time to tell the boss you've got cholera and take a month off. Let the plants die, leave the dog with a 55 gallon barrel of kibble and work around the clock for Autodesk. If you have skills as a programmer, use them—if you need any resources, machines, peripherals, software tools, coercion, let me know and *they will be provided*. We must get our OEM conversions done and our wish list—the entire wish list—implemented.

If you have management skills, offer to take over AutoCAD project management tasks—there are many. If your skills are in general management, finish our business plan, write job descriptions and interview people, and help us budget the spending of that wad of money we need to spend to make Autodesk a winner.

Or, when your company is in its time of dire need, you can go on putting in your hours faithfully, or maybe shaving a bit here and there. There are a lot of good movies around this summer, and it would be a pain to miss one. This is how you change from being a principal in the company to a stockholder.

Move to Mill Valley for a month (all expenses paid). Spend a week in the center of the cyclone at 150 Shoreline. Help write copy for our advertising campaign. Coordinate the ad agency design efforts. Help Mike Ford return calls to prospects.

Write our tutorial on AutoCAD. Work with the people we've already identified who want to develop shape libraries. Help me read these tons of competitive product manuals and design the wish list features. Get the Sony running, "I hear they sell well".

The people who make the all-out effort at this pivotal time will be the people who form the cadre who will run the company, if successful, for the next several decades. Those who "can't", "won't".

The management of this company will exercise its legal obligation to deploy the assets of the company so as to best ensure the success of the company. This means that we will spend what we feel we need to purchase outside services for those items we cannot do in-house. We will bring new employees into the company as we see fit, with compensation packages tailored to their individual requirements. This may include stock options granted at the current market value of the stock or above. To do otherwise would be to jeopardise the future of the company.

[143] By this, I was exhorting people to neglect their *real* jobs in order to work more for Autodesk. Most founders were still moonlighting for Autodesk at this point. The fact that, at this point, "job" didn't mean the work one did for Autodesk was a major contributor to the crisis that prompted this letter.

This is not the time to worry about your share of stock. This is not the time to agonise about how many options you got last time, or how many you will get the next. Today isn't the day for saying "we should" or "we ought to"; it's the day to say "I will", and then do it. Anybody who can spend any amount of time in recrimination or worry about anything other than our critical, immediate, problems, is not somebody working full time for this company. Let's defer the worries about conflicts of interest for a while. Let's not bitch about the other guy's royalty deal. Let's not say "I won't work with him" or "I won't work on that". Let's try to put a moratorium on "what's in it for me" reasoning.

Why? Because if we don't, nothing's going to be in it for anybody. Your investment in this company, your options, and your warrants, will be of no value except as wallpaper. All your work for the company will have been in vain. All the risks you took to join the company will have been for nothing.

Those who participate will know who they are. They can be assured of sharing in the prosperity of the company, if that is the result. Those who do not will know who they are. They will understand the difference between a "stockholder" and a "principal". For most people in this company there has never before been such a pivotal point in their business careers: the opportunity to exert oneself and gain the experience and track record to write your own ticket. Or to be an employee forever.

But it isn't all gloom and doom. The promise of this company is as bright as it ever has been. We are sitting in a window of opportunity, and if we can seize the moment, and act with a unity of purpose to make the company a winner, we can make our company one of the success stories of the 1980's. We have manufacturers breaking down our door to get us to put our product on their machines. We have the first product of our kind in the marketplace, and the best, and we have the expertise in house to add to our product all the things that the "big CAD" systems do.

Digital Research and Microsoft didn't succeed because they had access to technological breakthroughs. They made it because they had the audacity to put "things only the big computers can do" onto "toys", and then the business savvy to build on their initial success and become industry leaders. We've had the audacity. We've had the initial success.

We can do it. But only by mobilising the resources in the company fully to achieve what we must get done. And to do that, you must act. Now. Before you punch holes in this and file it in the Autodesk binder. Before you check the *TV Guide* to see what's on tonight. *Now!* Run to the window, throw it open, stick your head out and scream at the top of your lungs, "I'm going to make Autodesk a winner, and I'm going to work my ass off to get there". Then pick up the phone. Call me. Or Dan Drake, Mike Ford, John Kern, or Jack Stuppin. Then tomorrow we'll start building this company toward the next factor of 10 in growth.

Only you can do it. The full time people in this company are working at or beyond capacity. We need help, your help, or we will fail. And our company will fail. This is the moment. Seize it.

I never said it would be easy.

Sincerely,

John Walker

John Walker's Business Plan

The organisational problems alluded to in the notes from the June meeting continued to plague the company. It was clear that we could no longer function free-form. We needed a structure to organise all of the tasks that were underway, and a budget to help us deploy our growing, but still meager, financial resources.

I wrote this proposal in an attempt to specify a structure that approximated how the company was, in fact, already operating. It was never formally adopted, but what actually happened was not very different from what was envisioned herein. This is not a "business plan" in the venture capitalist sense; instead it was intended as a plan to develop the business, not raise money for it.

Autodesk, Inc. Business Plan

by John Walker — June 23, 1983
Revision 3

It is my feeling that the problems that beset this company are of such urgency that immediate action is required, with or without an agreed formal business plan. Herein I will discuss the actions I see as needed. I make no claim that I have prepared the requisite forecasts to back up the figures I use—they are pure seat of the pants numbers based on my gut feelings of where we are and what we can risk.

Organisation

I want to immediately reorganise the company in the following divisions:

- Marketing Division—Mike Ford, Manager
- Operations Division—John Kern, Manager
- Technical Division—Dan Drake, Manager

These divisions are coequal (listed in alphabetical order[144]) and all report to the president. Each division manager shall have the authority to disburse his budget (see below) as he sees fit. This budget may be used

[144] Do you detect any signs of friction?

for compensation of existing personnel to obtain additional work, or to bring in outside personnel. Approval of stock offerings to new personnel will be at the discretion of the board.

Budgeting

We have to start spending our money to establish this company in the marketplace. The following is my proposed monthly budgets for the above-designated divisions. These are based on seat of the pants "wing it" insight.

Marketing Division	$14,000
Operations Division	$10,500
Technical Division	$10,500

These figures were calculated on the basis of exhaustion of our in-the-bank capital in 4 months assuming no additional revenue from sales. The budget I propose would make me extremely unhappy were I the manager of each of the above divisions. Thus, I claim it is close to reality.

The above numbers also closely equal my estimate for the net revenue from sales if we assume that Sun-Flex takes as many systems as they have taken over the last three months, and that other sales stay equal. We can augment the above figures by spending capital or forecasting increasing sales, but I would be uncomfortable with an increase beyond 25%.

I assume that any increase in sales over the next 90 days will be used to increase the above budgets proportionally. I am open to the establishment of a separate customer support division, but as there is nobody to manage it currently, now isn't the time to do it.

Existing salaries are to be paid from the above budgets. The technical budget will pay for Greg and Duff's salaries, and the operations budget will cover John and Jane Kern, and Kathy Marcelius. The royalty payments to Mike Riddle and Mike Ford[145] are assumed to be subtracted directly from sales and are not included in these budget figures.

The technical budget assumes that we can continue to get the machines to develop on for free and that we can do effective work in that environment. If this is not correct, we would have to steal from the Operations or Marketing budgets for equipment.

The marketing budget allows for extensive advertising and additional personnel. If the advertising can be arranged on a co-op basis, or the personnel can be recruited on a commission basis, funds from this account may be reassigned to the others.

Priorities—Marketing Division

First priority for the marketing division is the recruitment of additional marketing personnel. Within the marketing division I place the receptionist/order-taker function, as well as the customer support function. We

[145] Actually, Mike's payments were the commission on sales agreed to in January, 1983.

need to have our phone answered reliably within 3 rings within business hours. We need to have a trained customer person on call at all times. We must authorise expenditures to achieve these goals. Further, we need an additional full-time marketing person, whether reporting to Mike Ford or coequal with a defined territory. Ideally we can recruit this person on a stock+commission basis. Next, we basically need to take the money left over and turn it over to Chuck Victory with the instructions "create us an ad campaign". We don't have the resources internally to do this advertising function, so we have to buy it, regardless of the price. Perhaps upon her return, Roxie Walker could be attached to the marketing department and given responsibility for development and execution of the advertising, show, and promotion campaign, a function she has done for us in the past. We must establish ourselves as the industry leader, with brochures, advertisements, and point of sale material that confirms our position.

Priorities—Operations Division

The operations division is immediately authorised to obtain a larger office facility and telephone and support resources as required from the assigned budget. The personnel to handle the additional telephone lines are understood to come from the Marketing budget.

Priorities—Technical Division

The Technical Division manager is empowered to spend his budget as he sees fit to accomplish the technical goals of the company. These include the completion of the "wish list" development goals, and the conversions to various hardware products. The manager of the Technical Division is empowered to hire outside personnel or to cut special deals with Autodesk shareholders as required to complete the development agenda.

Technical Project Management

We need a way to go outside for technical tasks without causing dissension and salary bidding wars internally. Perhaps the answer might be to simply take the wish list (or conversion list), and price each item on it (along, of course, with a completion date based on the estimation of the project manager). These items would be offered internally to the stockholders, who could sign up to do them and receive payment according to the scale. (This pay would presumably be in lieu of salary for those on salary.) The residue of unassigned projects could, at the discretion of the technical division manager, and constrained by the budget, be contracted out to outside programmers at the same price scale.

Reporting to the technical project manager are the AutoCAD-86 product manager (Dan Drake) and the AutoCAD-80 product manager (John Walker). The technical project manager allocates resources among the projects.

Salary Scales

The managers of the respective divisions must be in a position to offer stockholders full-time employment, or to bring in additional people. These decisions must be based on sound business practices based on the budget available to get the job done. I propose that each manager strike the best deal possible with each person, including compensation in stock or options if that seems acceptable (all stock offers having to be approved by the board, per my reading of our bylaws). I personally will take the heat from the existing employee/shareholders and handle any requisite renegotiations with them.

Proposed Autodesk Organization

The manpower and management crisis that became manifest in June of 1983 prompted many people in the company to suggest means for resolving it. Kern Sibbald submitted the following plan for reorganising the company and making the leap directly from our loosely-coupled mode of operation to full-time professional management.

This plan essentially anticipated all of the development in the company's management through November 1986. One can only wonder how much more smooth the company's development would have been and how much more success would have accrued had we found the courage to take these steps when Kern proposed them in 1983 rather than piecemeal over the next three and a half years.

Proposed Autodesk Organization

by Kern Sibbald — July 4, 1983

Recently there have been several Autodesk Business Plans or Organizational structures submitted. These are a very necessary part of a solid business, and I am in general agreement with the plans submitted to date. I have taken those plans, carefully considered the ideas in them, and added a number of ideas of my own. Only by being completely honest with you do I feel that I can present my concerns of the company direction and my proposed solutions, so I apologize in advance to those who may feel offended with my remarks.

Problems

I agree with John Walker that our company is beset with serious problems. However, I would say that it is a result of our spectacular success rather than a failing of our company. AutoCAD is very successful, and Autodesk has tremendous people skills and resources to draw on, but we are not making effective use of the resources that we have. For example, neither of our two most valuable resources, John Walker and Dan Drake, are available full time for working on AutoCAD. John seems to be spending much of his time on trivial matters and has complained that he cannot get off the telephone. His Marinchip business continues to occupy some of his valuable time. The situation for Dan seems to be similar. If we could spring John and Dan free from nontechnical work and outside pressures, they would produce four times the output that we could gain by hiring any other two individuals.

Out of necessity, we have begun to hire people to fill critical needs. But, we must attempt to hire those who have contributed the most to the company first. At the same time we must take care to define our needs in

terms of job descriptions and only then hire someone to fill that job rather than make a job to fit a particular individual. Once a person is on board, we should make every effort to fit the work to the person rather than the person to the job. This will produce much happier and more productive employees. But fitting the job to the person should not be done when initially hiring.

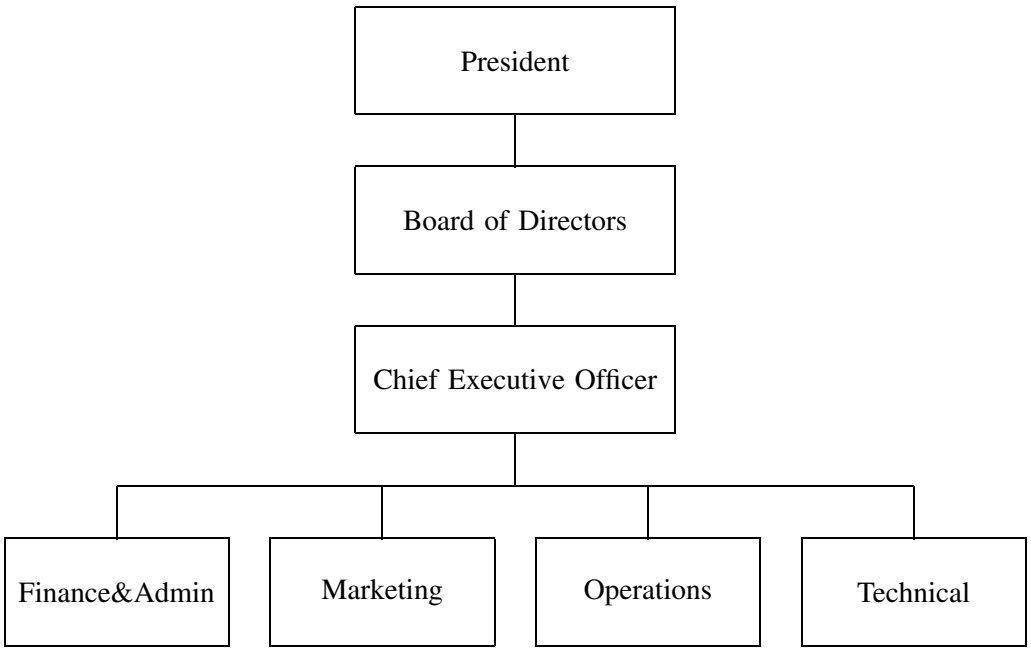
Because of the constant bickering, I often feel that we have little sense of purpose and direction. This is in part due to our unusual problems of wide geographic dispersion of our founders and primary work force. Consequently, to survive, our company will need to have communications that are better than most companies, yet our communications among each other are far from adequate even within a less geographically spread company. I view our problems as a crisis in management. This company owes its existence and the development of its only product to John, and we should never forget that. John Walker is an absolute genius in computers. He has led us through the early stages of the company, but now more than ever, his skills are needed urgently in enhancing our existing software and developing new software. His abilities are hampered by the trivia and details of a rapidly growing company, and we are deprived of his most needed skills.

Solutions

To develop into a major multi-million dollar company that we all envision, Autodesk needs additional management skills. I have prepared a proposed corporate structure and preliminary job descriptions for five key individuals that I feel will address the problems stated above. Most of these positions are already filled with qualified individuals; for other positions there are individuals within Autodesk who are qualified; and there is probably at least one position that will have to be filled by someone outside Autodesk. This structure may seem like a big unnecessary leap from our current structure to some of you. I agree, but I cannot imagine how we will get the job of running this company done without it. Each and every one of these positions is essential. Judging by the response to the previous proposals, I expect the only objections to my proposal to be to the two new positions (CEO and Finance & Administration). Consequently, I will discuss only those positions but briefly. First, the CEO position is needed primarily to offload John and Dan. With the structure that I have proposed with the CEO reporting to the board, John and Dan will continue their leadership role in the company but be freed of the daily trivia. His primary tasks would be to promote our company philosophies, improve our company communications, and offload John and Dan by handling the daily running of the company. The Finance & Administration position is equally important since his primary tasks would be to get the administrative parts of the company going (accounting, budgeting, policies, etc.), and more importantly to prepare our business plan and financing. These are essential if we are going to get venture capital or go public and sell stock (the only way we are going to become millionaires). Although, the F&A job may not be critical today, I strongly recommend that we find someone now so that he will be thoroughly familiar with the company when his services are critical.

Proposed Structure

The structure that I am proposing looks like this:



President	John Walker
Board of Directors	John Walker, Dan Drake, Keith Marcelius, Mike Ford
Chief Executive Officer	(open)
Finance & Administration	(open)
Marketing	Mike Ford
Operations	John Kern
Technical	(open)

Recommendations

1. That the board of directors approve the company structure and job descriptions presented here.

2. The board of directors proceed to immediately fill the CEO position and turn the job of finding suitable candidates for the other two positions over to the CEO, who will in turn make recommendations for approval by the board.

3. That none of the five senior management serve on the board of directors. This would require Mike Ford to resign from the board and a replacement be found. I recommend that we obtain someone who is not a shareholder in Autodesk (possibly a banker).

4. Provide economic incentives for John Walker and Dan Drake to close Marinchip.

5. Reduce our need for part time employees by replacing them with full time employees over the next several months.

As a final note, the job descriptions attached should be considered preliminary since I have probably not included all functions, and some functions listed under one manager may be appropriate under another.

Job Descriptions

Chief Executive Officer (CEO)

Job Title: CEO
Grade:
Full time

Reports to: Board of Directors

Job Description:

Acts as chief executive officer of Autodesk, Inc. and as such is responsible for overall health of Autodesk, Inc. Under broad operating guidelines from the board of directors assumes the full responsibility for keeping Autodesk, Inc. a profitable corporation.

Job Responsibilities:

Plans, directs, coordinates, and controls the daily operation of Autodesk through the four division managers. Exercises the responsibility for preparation of all Autodesk budgets, submits these budgets to the board of directors for approval, and monitors expenditures against the budget. Directs the development of and approves standards and procedures. Responsible for development of Autodesk personnel policies and obtaining approval from the board of directors. Provides board of directors with monthly status report and detailed quarterly financial reports. Assumes responsibility for all aspects of daily operation of Autodesk including hiring, firing of employees, determination of employees' salaries within salary guidelines approved by the board, organizational structure, staffing within approved budgets, approving expense reports. Promotes the Autodesk company philosophy of "Excellence in Computer Software".

Job Qualifications:

- Prior management experience, preferably in a high-technology startup company.
- Demonstrated written and oral communication skills.
- Ability to provide leadership to a diverse group of people.
- Ability to handle multiple simultaneous tasks and to function well under pressure.
- Working knowledge of developing, maintaining and supporting computer software.
- Experience formulating objectives, standards, and procedures.
- Knowledge of negotiation and administration of contracts and legal aspects of a corporation.
- Working knowledge and experience developing budgets and using cost control techniques.
- Experience managing and evaluating technical and supervisory personnel in a data processing environment.

Manager of Finance and Administration Division

Job Title: Finance and Administration manager
Grade:
Full time

Reports to: CEO

Job Description:

Responsible for all aspects of Autodesk finance and administration. Reports on a daily basis to CEO and maintains frequent contact with other division managers.

Job Responsibilities:

Responsible for developing and implementing financial and administrative procedures such as: payroll, bookkeeping, insurance, budget and cost control, personnel policies, accounting functions, procurement procedures, contracts, inventory management and control procedures, security of facilities, legal protection of Autodesk software rights, financial reporting, strategic planning, obtaining appropriate financing, developing the Autodesk business plan. Develops budgeting methodology and aids other division managers in preparing their budgets and quarterly reports. Consolidates budgets from division managers on a quarterly basis or more often as needed for submittal to CEO. Monitors division managers actual cost and recoveries versus budgeted cost and recoveries and prepares monthly reports for submittal to CEO. Prepares comprehensive Autodesk financial reports quarterly. Ensures that taxes and other payments are made in a timely manner to all government agencies. Responsible for maintaining and publishing all software documentation in coordination with the other division managers. Responsible for annual salary survey to ensure that Autodesk compensation structure is competitive with the computer software industry. Publishes monthly report to stockholders. Provides adequate written communication to keep all Autodesk employees and stockholders appropriately informed.

Job Qualifications:

- Prior experience developing budgets.
- Knowledge of computerized cost control and accounting systems.
- Demonstrated written and oral communication skills.
- Knowledge of negotiation and administration of contracts and legal documents.
- Experience formulating objectives, standards, and procedures.

Manager of Marketing Division

Job Title: Marketing Manager
Grade:
Full time

Reports to: CEO

Job Description:

Responsible for all aspects of Autodesk sales and marketing activities. Reports on a daily basis to CEO and maintains frequent contact with other division managers.

Job Responsibilities:

Responsible for sales, advertising, customer support, marketing research. Prepares marketing budget on a quarterly basis or more often as need for submittal to CEO. Prepares monthly sales forecast and identifies deviations from approved budget. Prepares sales and advertising plans and submits to CEO for approval. Maintains close contact with operations manager to coordinate planned sales, inventory, mass mailings, etc. Hires, trains, and provides supervision to telephone sales employees. Develops and submits sales incentive programs to CEO for approval. Maintains close contact with manager of technical division to coordinate release of new software and to provide customer satisfaction and quality assurance feedback. Responsible for providing support for end users. Recruits, trains, and supports dealers for Autodesk software. Develops OEM contacts with hardware manufacturers and software houses for Autodesk software.

Job Qualifications:

- Prior marketing experience, in a high-technology company, preferably in a startup situation.
- Demonstrated written and oral communication skills.
- Demonstrated ability to create innovative sales incentive programs.
- Demonstrated ability to hire and manage a geographically diverse sales organization.
- Demonstrated ability to make contacts with large hardware companies that are likely hosts for Autodesk software.

Manager of Operations Division

Job Title: Operations Manager
Grade:
Full time

Reports to: CEO

Job Description:

Responsible for all aspects of Autodesk manufacturing and operations. Reports on a daily basis to CEO and maintains frequent contact with other division managers.

Job Responsibilities:

Responsible for manufacturing, shipping, receiving, distribution, courier service, mass mailings, bingo card responses, maintaining a customer database, facilities, and inventory control. Prepares operations budget on a quarterly basis or more often as need for submittal to CEO. Prepares a monthly status report identifying deviations from approved budget. Maintains close contact with marketing manager and technical manager to coordinate planned sales, inventory, mass mailings, etc. Hires, trains, and provides supervision to office employees. Maintains close contact with manager of technical division to coordinate release of new software and appropriate inventory levels. Responsible for stocking and providing supplies to all Autodesk employees. Orders all equipment and supplies for Autodesk and Autodesk employees. Responsible for all incoming mail and maintaining answering and message services for all Autodesk employees. Responsible for paying all invoices. Recruits, trains, and supervises all office employees.

Job Qualifications:

- Demonstrated written and oral communication skills.
- Demonstrated initiative to solve operational and manufacturing problems.
- Prior experience managing an office environment.
- Ability to maintain good working relations with people of diverse skills.
- Ability to work well under critical deadline presures.

Manager of Technical Division

Job Title: Technical Manager
Grade:
Full time

Reports to: CEO

Job Description:

Responsible for all aspects of technical software development and support. Reports on a daily basis to CEO and maintains frequent contact with other division managers.

Job Responsibilities:

Responsible for all aspects of Autodesk software products. This includes: software development, preliminary documentation, source code control, quality assurance, software maintenance, error tracking and reporting, software standards and guidelines, hardware support, software release, implementation of Autodesk software on new hardware, technical user support, and new software product research and development. Prepares technical division budget on a quarterly basis or more often as needed for submittal to CEO. Prepares a monthly status report identifying deviations from approved budget and progress on software development projects. Maintains close contact with other division managers to coordinate release of new software and documentation.

Job Qualifications:

- Demonstrated written and oral communication skills.
- Prior successful experience managing large software development projects.
- Demonstrated ability to manage technical personnel.

July 1983 Meeting

Sales continued to build and the July meeting reviewed the progress of a company clearly on the rise. Focus had shifted entirely to AutoCAD, more and more founders were coming onto the payroll, and the negotiation of sales and marketing agreements came to the fore as we tried to secure a stable base of revenue without compromising our freedom of action in the future.

This meeting marked the end of the original style of managing the company through monthly meetings of founders. As more and more new employees were hired, sales volume increased, and the action came to centre more and more on the office, the key management meeting came to be the weekly status meeting in the office. This was the last monthly founders' meeting. The next time the founders gathered to review the progress of the company was more than two years later, after the public offering, on Moon Day, 1985.

Autodesk, Inc.
July Monthly Meeting (July 9, 1983)

by Kern Sibbald

Attendees

The meeting was held at Jack Stuppin's house at 1:00 P.M. The following people attended: Dave Kalish, Hal Royaltey, Jack Stuppin, Kern Sibbald, Mike Ford, John Walker, Roxie Walker, Keith Marcelius, Dan Drake, Greg Lutz, Mauri Laitinen, John Kern.

Before the meeting began there was much discussion about whether or not to hire a copyright lawyer. The discussion centered on how much it was going to cost, and most people seemed to dislike an open-ended deal. We agreed to hire the lawyer (I didn't get his name) to do a copyright or trademark search on AutoCAD. We also discussed hiring him to review a contract with AlphaMerics. The contract involves us gaining access to their extensive set of symbol libraries.[146] During the meeting, Jack and Mike left to review this contract with the lawyer. AlphaMerics has an exclusive with NEC to package all NEC OEM hardware for CAD. We discussed AlphaMerics purchasing 200 AutoCADs in a year, 30% in 4 months starting Sept 1 at $450.00 each with a minimum of 10 per month thereafter.

[146] It was the prospect of supporting these symbol libraries that prompted us to include assorted odd capabilities in AutoCAD's **SHAPE** definitions, including "fractional arcs".

Summary of Management Meeting: Dan Drake

The books for last year are closed (see enclosed summary). Keith has accepted the AutoCAD project management position. We have hired Jack O'Shea as a full-time telephone salesman working for Mike Ford. He will also be working on filling out our dealer network. He is a retired police officer and ex-draftsman. We are looking for a larger office so that we can move the sales out of Mike Ford's house and have room for demos and possibly meetings.

Financial Report: John Walker

Due to the amount of financial activity lately, the report is approximate:

Savings	$77,000
Checking	25,000
We owe:	
Mike Ford about	$1,000
Mike Riddle about	$9,280

See attached sales report for more details.

Marketing Report: Mike Ford

Our sales are very closely tracking the curve that he forecast in January but we are about a month behind. SunFlex is now accounting for only about one half of our sales (this is clearly shown on the sales summary sheet[147]). Mike forecasts that the rapid growth trend will continue. We now have several competitors but at least in the case of P-CAD it is not yet for sale. Rik[148] of Mike's staff is writing an article on CAD. Four people from TI visited us in our office in Mill Valley and all went well. Two were from corporate and two from the engineering workstation project. They seem to be very impressed and we were asked to purchase a TI and they agreed they would purchase enough AutoCADs to reimburse us. Zenith will have a machine here Monday. Heath has decided they want about 250 AutoCADs in the next year. They put a fire under Zenith. Heath wants it on a Zenith so they can expose it to their salesmen before the end of July. Note: on ordering machines, etc., request that the machines be shipped directly to your house then give John Kern a call so he can pay the invoice when it comes. Many shipments for AutoCAD members have been coming to the office and that results in delays getting the equipment to you. Mike Ford and Duff Kurland went to NCGA[149] for 4 days—it was a high class show with all the big CAD guys (Auto-Trol, Computervision, etc.). There were about 22,000 attendees. It was worth being there since only about 25% were end users, the rest were dealers, manufacturers, and educators. There were 5 competing CAD packages at the show. P-CAD, MARS ($14K including hardware), Microcomp ($6K for software), Bausch and Lomb ($29K), and Summagraphics, (poor

[147] Lost in the mists of time.
[148] Jadrnicek.
[149] National Computer Graphics Association trade show.

notes here). Nine manufacturers came to see us. Mike Ford recommends that we lend an 8087 to Neil Zackery who is supposedly going to write an article about AutoCAD. Greg Lutz agreed to buy the 8087. Tektronix may want 100 ACAD's.

More Marketing: John Walker

We are about to conclude an exclusive distributorship with Jamal[150] for ACAD-80 except for the Sony. Beginning Monday John Owens will be dropped as a distributor. Hopefully, Jamal will pick up support for Owens' customers. Jamal has sold 18 AutoCAD's to date. We are trying to approach Microsoft with some kind of graphics deal. "After the great Digital Research ripoff" where they picked our brains—cuz they are working on a micro based CAD,[151] they are now trying to exchange GSX for an AutoCAD. "No way". We can get a Corona PC if we want from a dealer. We have an Otrona in-house. Mike Ford suggested that we rename the company AutoCAD to avoid confusion. No one was really very enthusiastic but everyone did agree it would be much better to have the AutoCAD name really large on our display. Mike wants someone to go to SIGGRAPH.

International Sales

We have received a transfer of $2,000 from Lars but are not sure what it is for. There are still 15 invoices outstanding.

Corporate Status Report

The incentive stock options, John Kern's buying into the company (the shares we were going to sell to Jamal), and a request to split the stock ten-to-one are all in the hands of the lawyers. We may have to request stockholder agreement on the share splitting. Everyone at the meeting agreed that it was a good idea.[152]

Operational Report: John Kern

John is trying to keep the inventory low so we will be positioned to use the new generic manual when it is ready.[153] He is giving two day turnaround on CP/M systems[154] and one day on the other systems.

[150] Jamal Munshi, of MOMS Computing, our first customer.

[151] They weren't.

[152] This was the first stock split. It reduced the initial nominal value of the shares from $1 to $0.10.

[153] Up to this time, we had a different manual for the IBM and Victor versions of AutoCAD-86. The "generic" manual inaugurated the division into an AutoCAD user manual and a machine-specific installation guide.

[154] Because John Walker had to make each one to order.

AutoCAD-86 Report: Keith

We hope to release the MS-DOS Victor with the HP driver on the 18th. The configurator release is set for July 29. We have a feature freeze in effect that roughly agrees with the current state of Duff's generic manual. We discussed manuals some here. Roxie is going to try to get the next one so it will lay a bit flatter. After some discussion on various bindings and size, everyone agreed to keep the same manual size and binding. We would like to schedule major feature releases every 60 days.[155] The manual price is now $35.00 because SunFlex insists on a discount and we won't lower their price below the current $25.00. We agreed to continue to sell manuals at $25.00 if customers ask at that price since we have advertised it so much and don't have price change disclaimers on our literature. The single-screen IBM is working and almost ready for release (1 week).[156]

AutoCAD-80 Report: John

AutoCAD-80 works on the Sony. It is an absolutely beautiful machine according to John. They immediately sent him everything he asked for. One can get a complete system with a color printer? and a lot of other stuff for $4,100.00.

Miscellaneous

John wants to hold the Autodesk meeting quarterly rather than monthly. There seemed to be general agreement. However, everyone agreed that we must solve the communications problem first. That is we must somehow continue to write down information on that is happening in the company and send it to everyone. John suggested that Kern continue to put out a newsletter and that he can get input every month from the division managers. There was also some discussion of holding the weekly business meetings on Fridays so that more people can attend. Apologies to Lars for misspelling his name in the last month's notes.[157] It is Lars Åke Moureau. David Kalish has received no bug reports.

Personnel

We now have five full-time paid employees: John Kern, Keith Marcelius, Duff Kurland, Greg Lutz, and Jack O'Shea. There are two half-time employees: Kathy Marcelius and Jane Kern. John proposed that Roxie Walker join the company as Assistant to the ACAD-86 manager (Keith) and as our Arts director—probably full-time. Dan Drake may be employed full-time by Autodesk retroactive to July 1.[158]

[155] We still would. We couldn't then, and we can't now.

[156] Up to this time, our IBM CGA version required both the CGA and the monochrome adaptor and two monitors. The IBM single-screen permitted a CGA-only configuration.

[157] Fixed in the mix.

[158] He wasn't.

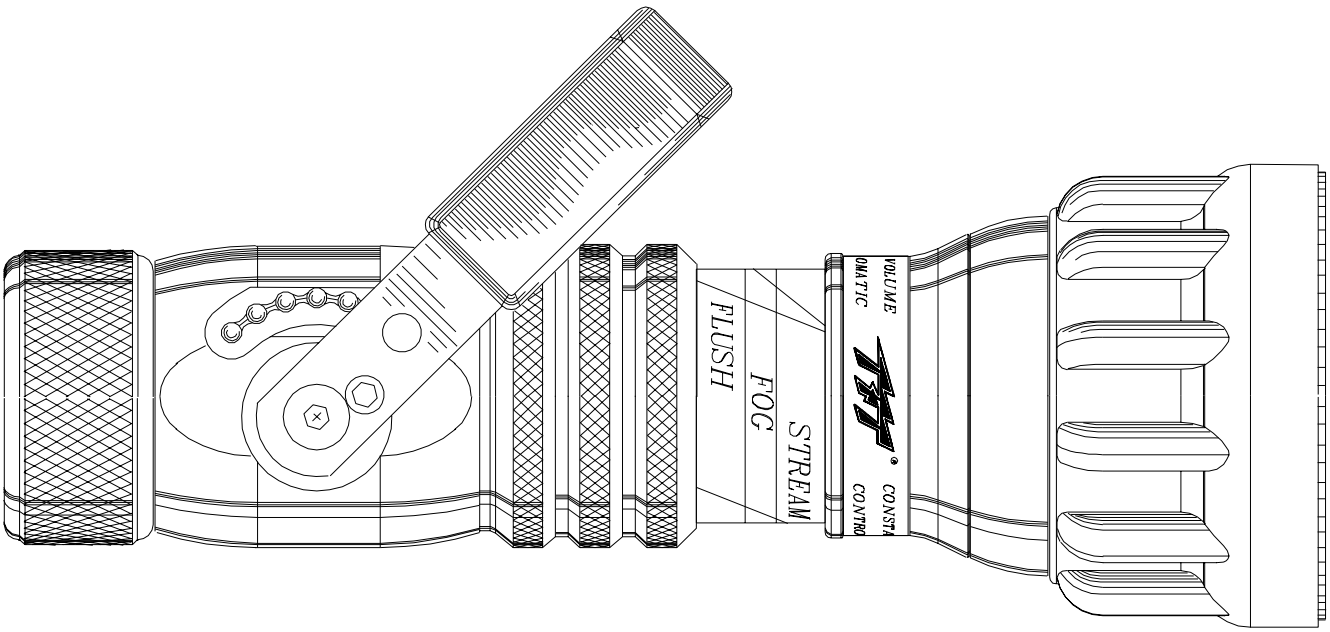
Equipment List

Very few of you have sent me your hardware configuration so I am giving up on publishing a list of what we have. Thanks to those who did: Duff Kurland, Richard Handyside, Rudolf Künzli, and David Kalish.

Kern

The most famous AutoCAD drawing of all: Don Strimbu of Task Force TIPS in Indiana created this drawing with a very early release of AutoCAD. It's impossible to describe the impact this drawing had when he sent it to us: it was the first really complicated drawing that had been done with AutoCAD, and Don's cleverness in using block scaling to simulate perspective on the text and to mirror parts of the nozzle astounded us all.

We immediately began to use the nozzle as a standard timing test for machines. I joked at the time that someday people would talk about "nozzle standard units" instead of Whetstones—never did I think that would come to pass. The nozzle drawing and the actual nozzle were featured in our first four colour two-page advertisement, which ran in Scientific American *in September 1984.*



Low Rent 3D

Here's the first concrete proposal to add three-dimensional capability to AutoCAD, from September of 1983*. We did not embark on the mad rush to 3D that this proposal urged. Instead, we deferred implementation of 3D until AutoCAD 2.1, which we previewed at COMDEX in 1984, 14 months after this paper was written, and shipped, and shipped again, and shipped yet again, in May of 1985, 17 months later. The 3D Level 1 that we shipped in version 2.1 was, in some ways, more limited than that proposed herein. It would be interesting to know how history would have unfolded if we'd done this.*

Low Rent 3D

Proposal by John Walker — September 5, 1983

If Autodesk is to prosper, it must continually enhance its products and introduce new products. This becomes especially true as other people introduce competitive products. To maintain market share and keep the price up, adding capabilities to the package is the foremost technical contribution that can be made.

With the features scheduled for the 1.40 release, plus the items which we hope to have in 1.5[159] (notably dashed lines, double walls, and some form of attribute collection and dissemination), we will have accomplished most of the goals inherent in a 2D drafting system.

Many of our competitors (MCS, Nelson Johnson, ESC), have or will be introducing 3D packages around the time of COMDEX. If we do not have a credible response to queries about 3D, we may be in trouble selling our package. While all drafting is 2D, and almost all users will spend all their time with AutoCAD working in 2D mode, 3D is important more from a marketing perception standpoint than a technical one.

First, there is the natural tendency to evaluate a package from the features it has, and a package limited to 2D cannot look as good on a cursory examination as a 3D package. Second, many companies will reason, "look, all we need is 2D today, but who knows about tomorrow; we better buy a package that has 3D just in case rather than get stuck with a dinosaur". Note that this applies even if 3D is an expensive option that they don't buy at all: just knowing it's there may clinch the sale. Third, 3D demos beautifully and is an extraordinary sales tool. The impact of rotating an object in 3 space at COMDEX is many times that of zooming in on a flat drawing.

What I advocate here is a particular way of adding 3D to AutoCAD. I think that (from my very limited knowledge of the 86 version internals,[160] remember) it can be done in a limited time without disrupting the

[159] This version was eventually released as AutoCAD 2.0 in October of 1984.

[160] At the time I wrote this, I was still working exclusively on the Z-80 PL/I version.

other development in progress concurrently. This is a prime consideration—we cannot afford to stop other development while somebody rips the package apart and changes everything for 3D. The method imposes restrictions on the use of 3D which I feel are acceptable, and puts no large barriers in the way of removing the restrictions in the future. Virtually none of the work done in installing this package will have to be discarded when we go to full 3D.

OK, loudmouth, so what is it you're proposing anyway?

The idea for Low-rent 3D is to allow 3D representation of objects but restricting these objects to lie in one or more planes in 3 space. The planes are completely arbitrary, and are not restricted in any way. These planes map into our layers. For each layer you get to define its origin point (e.g., where the coordinates 0,0 on that layer are in the master X,Y,Z space) and the orientation of the plane (by 2 other points, angles off the axes or whatever).

When you're drawing, you're simply adding entities to that layer in terms of the coordinates in that layer, and that's how they go into the drawing database. *There are no changes to* **EACQ** *or entity-generating commands.*

When **EREGEN** generates an entity, it generates vectors as currently done. It passes to **CLIP** the endpoints of a vector to draw in the plane of residence of the entity. *There are no changes in* **EREGEN**.

When **CLIP** receives the endpoints of a vector, it maps them from the coordinates in their layer of origin to their true 3 space coordinates. **CLIP** is of course rewritten to do a 3D clip and perspective (or isometric or whatever) transformation onto the screen. I suggest that at the same time we enhance **CLIP** to support multiple views on the screen—merely additional projections from a table.

Since everything is ultimately turned into vectors in a flat plane (the screen), the refresh file doesn't change, and you can pick items on the screen as always. *There are no changes in the refresh file, entity selection, or* **EID***-related commands.*

So, basically what we're changing here is this: We add a plane location for each layer in the drawing. We provide a command to let you specify (and move) this plane (probably heavily oriented to things like "I want a plane just like that one but with Y 10 greater", but maybe with a different syntax). This same plane specification mechanism is used to specify the viewing (image or screen) plane. For the viewing plane, we also need to specify a clip depth and transformation (perspective, isometric, how many vanishing points and where, etc.). We rewrite **CLIP** to, from real endpoints and a layer number, generate the true 3D coordinates and clip and project that vector on the viewing plane. We enhance **GRID** and cursor tracking from the digitiser to map through the transformation so that movement in the entry plane displays correctly on the screen.

And that's it.

Now notice just how modular this all is. We should be able to whip it off and integrate with little more trouble than any of our other development projects. And with nothing like the grief of "Well, we start by changing all the entities, then. . . "

But will it be useful? Yes, I think so. Look at most of the 3D demos you see in the literature or in our competitors' handouts. Think about drawing them in a system like this, and I think you'll agree (I exclude solid modeling systems, of course). Most 3D objects are built up of planar pieces, and I think can be represented in a system like this with little pain.

There is one addition I'd like to propose for this package at the inception: the concept of *extrusions*. With every layer would be an extrusion depth. If zero, this would have no effect. If nonzero, then when **CLIP**

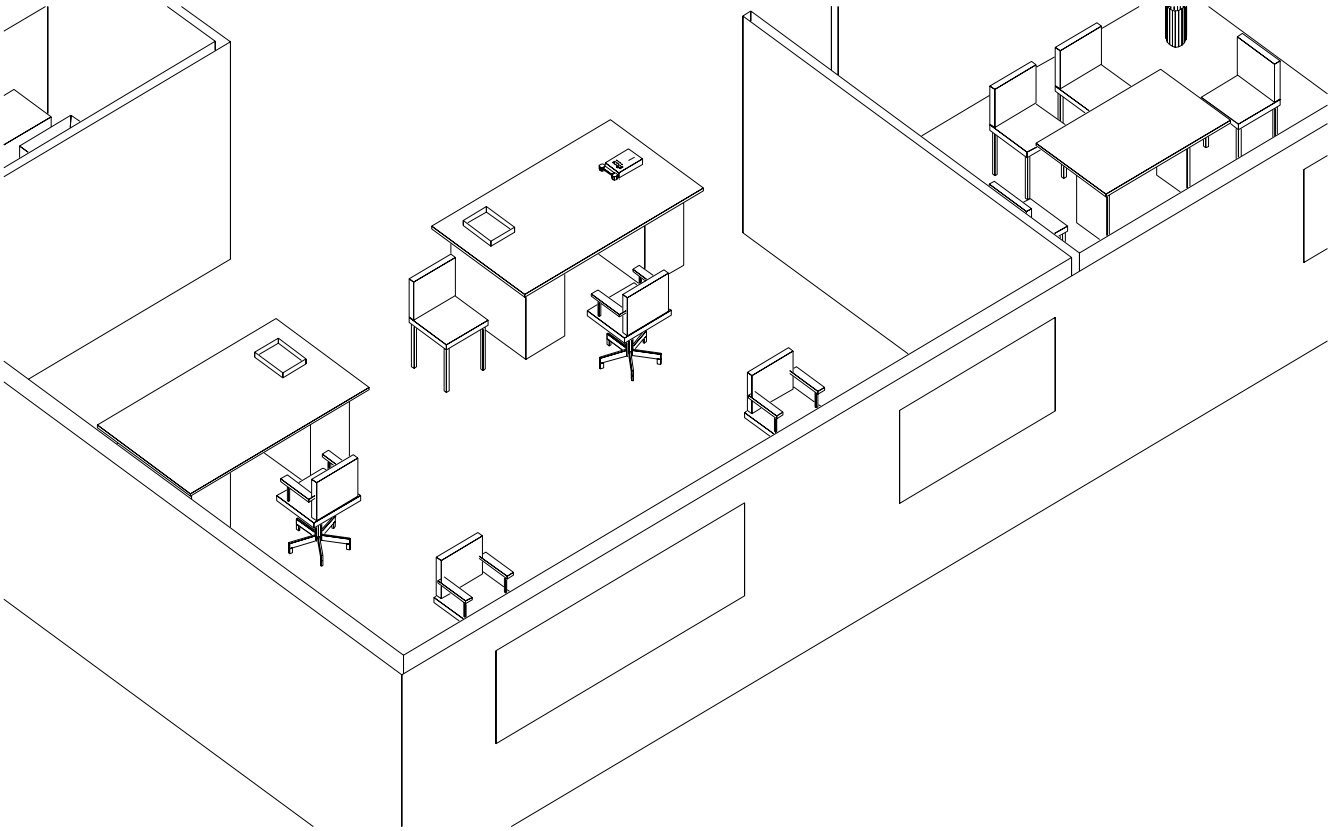
processed a vector, it would actually generate internally and clip 4 vectors, one in the plane, one in a parallel plane offset by the signed extrusion depth, and two connecting the two vectors in the planes. What for? Look at the gear on the cover of *PC World*. It's also neat for things like city skyline modeling. Obviously it can reduce the number of planes you need for lots of things.

Finally, I'd like to point out how little of this needs to be thrown away when we go to a full 3D (space curve wire frame) package. We add Z coordinates to all the entities and rip up **EREGEN** (e.g., I want this text written on that cylinder over there). But we end up passing 3 coordinate pairs to **CLIP**, projecting them, and viewing them, and all the code we do for that is used unchanged. We still need to specify viewing planes, so all our code for that gets reused. We will of course leave the low rent 3D package in, as it will, I think, serve as a friendly bridge between the second and third worlds.

It's my feeling that the magnitude of the task we're talking about here is comparable and very likely less than implementing the major additions packages we have underway or recently completed (crosshatching, circle generation optimisation, line types, fillets, etc.). And no other single addition will so well enhance the perceived value of AutoCAD, or its ability to sell itself at shows. Just think of the difference in our COMDEX literature of "Now with 3D" versus "Now with crosshatching and dashed lines". I think we should discuss this and if no major technical barriers are seen, go for it with a goal of introduction at COMDEX and shipment within 30 days thereafter. We can then let the market response tell us whether we should invest the work immediately for a full 3D package.

We can probably get away with a stiff price increment for this thing. If people pay $500 for the piddling dimensioning code, I think an extra $1200 for 3D is not inconceivable. I have absolutely no idea, however, how many we might sell or what the price/sales curve would look like. If Nelson Johnson is to be believed (!), though, hundreds of people are willing to spend $400 for a 3D package that is completely useless. And that with little advertising on just a few machines. So I think this can be a powerful revenue generator as well.

This drawing is a direct descendent of the first three-dimensional demo drawing that Duff Kurland put together for the introduction of 3D Level 1 at COMDEX in 1984.



Electric Malcolm

Whenever an Autodesk old-timer wants to intimidate a Kelvin-come-lately, the conversation always seems to turn to "Electric Malcolm". After a few veiled references to this legendary facility, the newcomer walks away slowly, shaking his head, and muttering something about "they told me this company was weird, but 'Electric Malcolm'? . . . naaah.". Well I promised to tell all, so here's the inside scoop on Electric Malcolm.

Malcolm McCullough was still studying architecture at UCLA when he took a summer job with Autodesk in 1983. Malcolm was the first person really talented in drawing to work for Autodesk, and his work with AutoCAD helped both by generating good sample drawings and by identifying the most important features needed in real professional drafting. The Golden Gate Bridge drawing that we used to feature so prominently in our advertising was drawn by Malcolm that summer.

We had hoped that Malcolm would be able to help us put together some form of scripted or video demo. When time began to run out, I implemented a transcript capture facility which would actually be able to record Malcolm creating a drawing. Then we could play it back at a trade show or in a dealer's showroom and show AutoCAD making a drawing in the hands of a master with the simple push of a button. Hence, "Electric Malcolm". The code was implemented in the CP/M-86 version of AutoCAD, but there was a stability freeze in effect prior to the release of Version 1.4, so the code was never integrated in the product. To this day, AutoCAD lacks this capability, although both AutoSketch and AutoShade support it for development testing purposes.

AutoCAD-86 Transcript Facility

Implementation Notes by John Walker
September 14, 1983

The attached disc contains the additions to AutoCAD to provide a crude transcript capture and replay facility. This code is provided for internal use only, and has several glaring shortcuts and deficiencies which are excusable only by the short time remaining before it must be pressed into service to prepare demos for NSS[161] and COMDEX, and the short time before our best AutoCAD expert departs for southern climes.

I have tried to implement this code in an extensible way, and will later suggest how existing transcripts prepared with this code may be painlessly converted into a more advanced version compatible with the 1.40 DIG changes and future plans.

[161] The National Software Show, now defunct.

First, let's look at how the mechanism works.

To make a transcript, at any time while AutoCAD is active, you may enter the command:

XSCR

You will be prompted with:

Transcript file:

and you should respond with a valid file name in the system you are running under. This file name may contain a drive letter, but must not contain a file type. A file type of "**XSC**" will be used for all transcripts. If a file with the given name already exists, it will be overwritten. Following the completion of the **XSCR** command, you will receive the "**Command:**" prompt, and henceforth every AutoCAD action you take will be recorded in the transcript file. That is, every keystroke on the console, every pick with the pointing device (whether in screen or tablet mode), and every menu pick, whether from the tablet menu or the screen.

To terminate the recording of the transcript, just enter the command:

XSCR

again. This will close the transcript file and turn off recording. The transcript file will also be automatically closed out when a command is entered which leaves the Drawing Editor (**END** or **QUIT**). But remember, if you use an **END** or **QUIT** in a transcript, it will be recorded and will take effect when the transcript is later used, so be sure this is what you wish.

Once a transcript file has been recorded with the **XSCR** command, it may be replayed at any time with the command:

RPLY

This command will prompt you:

Transcript file:

and you should respond with the file name of a previously recorded transcript. As with the **XSCR** command, the file type of "**XSC**" is assumed and should not be specified. The transcript will then be fed to AutoCAD as it was initially entered. If the transcript was terminated with an **XSCR** command, that command will display at the end, but will be ignored. If the transcript does not terminate the Drawing Editor, control will return to the console at the end of the transcript. Transcripts may not be aborted. (This isn't hard to fix.) Transcripts have meaning only within the Drawing Editor. Unlike **SCRIPT** files, they cannot be used to feed commands to the main menu, configurator, or plot modules. Note that you are free, however, to make composite demos with scripts which use the **RPLY** command after calling the Drawing Editor.

In using transcripts to prepare demos, it is of the utmost importance that you remember to save the precise initial environment which obtained when the transcript was captured. That includes the original drawing file (beware of making any changes, even of view, before starting the transcript), the menu file(s) in effect, all **LOAD** and **INSERT** files, and the same display hardware (since digitiser samples are converted into screen coordinates). A transcript is simply a logical baboon typing from a list of characters and moving the cursor to where it was on the screen before—if you change the environment, the baboon will just keep on typing with nary a giggle at the devastation which ensues.

Consequently, the wise transcript maker saves the entire disc set before the transcript capture, then makes the transcript, then sets up a demo script incorporating the backup disc set and makes sure the demo process isn't destructive of the initial information on the disc.

There is no way to edit or concatenate transcript files. Zip. Nor is there any reasonable way to convert a transcript from one machine to run on another, or to update it for a new version of AutoCAD. However, this is not as bad as you might think. We are making changes in the interface between DIG and the people who call it which will rationalise the way tablet mode and handling of screen pointing work. These changes will have a major impact on transcript capture, and will allow us to much more easily turn a transcript into a **SCRIPT** file which can be edited with a text editor. This is really what we want, so it doesn't make sense to make a large investment in a 1.30 base transcript mechanism now. But based on timing, we gotta have something now, so this is it. When we get the new interface, I can gimmick DIG to read one of these old transcripts while writing a new style one, and then everything gets automatically converted (I hope).

The transcript code itself is very obvious. COMMAND is modified to recognise the **XSCR** and **RPLY** commands (which have clunky names because we have no intention of making this facility available to users), and to add the code which closes an open transcript on an **QUIT** or **END**. All transcript code in both COMMAND and DIG can be turned off by undefining TRANSCR, which is how we will normally ship AutoCAD.

The code to process the **XSCR** and **RPLY** commands is in the procedures with the same names in DIG. Note that the transcript file is paged; we wish no more disc I/O than necessary, because we may fill a buffer during user keyboard input. Both the capture and replay code is added to the procedure DIGITZ, and is obvious. The format of the transcript file is:

00–7F	Console character
F0 SX SX SY SY	Digitiser pick
F1 SX SX	Menu pick
F2 RX RX RY RY	Raw digitiser coordinates (tablet mode)
FF	End of file

Any questions this doesn't answer are as easily answered from the code as from a document about it. The raw X and Y coordinates are written first so that RAWX and RAWY are correct when the pick item follows. It is always generated regardless of tablet mode.

Writing of the transcript file doesn't adjust the disc full counter. That's because I'm too lazy to bother with it for internal code I'm going to rewrite anyway.

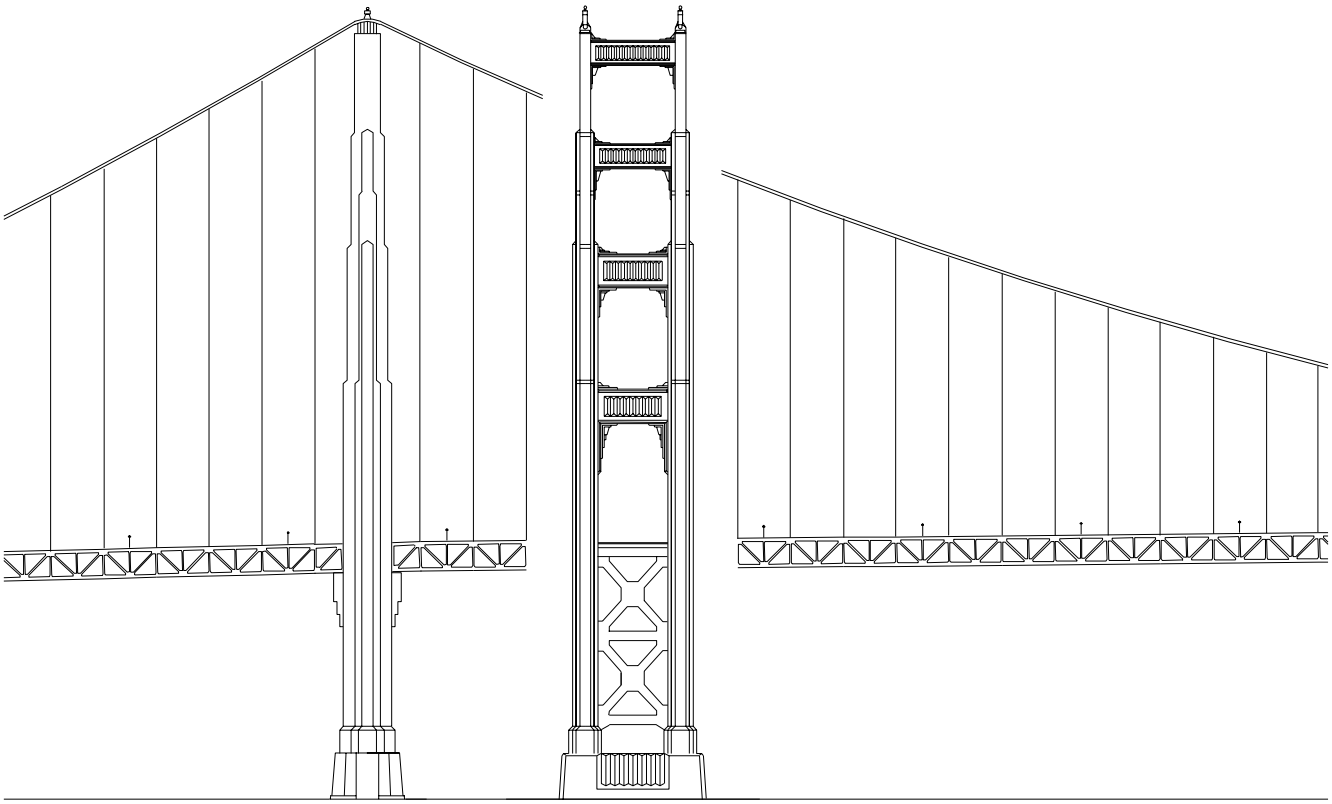
Freehand sketch material doesn't get captured in the transcript. This isn't particularly hard, but I skipped it because of lack of immediate need and to prevent code integration conflicts (as changes in SKETCH would be needed, and SKETCH in currently under integration into 1.40).

The management of the cursor in the replay code is no great shakes. It should really glide the cursor over to the point smoothly to simulate user movement. This would make the replay much better. We'll also have to put in **DELAY**s at strategic places after we get the translator to **SCRIPT** format working. All menu picks are turned into screen menu picks, with cycling through the NEXT box as required. Until we get the robot arm for the digitiser (or mouse with legs), this is the best way we can represent menu usage.

Anyway, the plan is that we retrojam this thing into 1.30 for the Victor and let Malcolm loose on it for the time we have left, then massage the material we collect in the free time after he goes. I'm sure we can come

up with a better mechanism in the future, but this one works and we can use it in the remaining time.

Here's the drawing of the Golden Gate Bridge that Malcolm McCullough did in the summer of 1983. His drawing shows the entire bridge and includes structural details beneath the panels. This view of part of the drawing is the one we featured most frequently in brochures. It's hard to appreciate what an achievement this drawing is without having used the primitive version of AutoCAD with which Malcolm drew it.



October 1983 Meeting

As decided at the July meeting, the full company meetings were now held quarterly rather than monthly. This letter announced the first of these quarterly meetings, which was combined with a special shareholders' meeting called to adopt a provision basically intended to extricate John Walker and Dan Drake from the disastrous financial consequences of buying their initial stock through Marinchip. Also, we believed ourselves to be hot on the trail of venture capital and we wanted to be able to adopt any provisions required by investors should a deal emerge.

This was the first general company meeting held at the company's offices. We finally had enough space to fit all of the stockholders in at once.

Autodesk, Inc.
150 Shoreline Highway, – Bldg B
Mill Valley, Ca. 94941
(415) 331–0356

Autodesk Quarterly Meeting

The October general meeting will be held at 1:00 on Sunday, October 9, at the company's main office:

150 Shoreline Highway, Building B
Mill Valley, California

The meeting has been moved to the second weekend of the month in order to accommodate arrangements for the special meeting of shareholders, for which an announcement accompanies this notice.

Note that the general meeting begins at 1:00, though the shareholders' meeting is scheduled for later.

The main topic of discussion will be our plans for marketing the product. Technically we are now well ahead of everybody, but competitors are beginning to sell products, and they're much better financed than we are. To avoid being swamped by their gigantic advertising budgets, we are now looking very seriously into getting outside money. This could be a large change in the organization of our business, and may require some official actions by the stockholders; we have called the special meeting so we can take action if necessary.

As noted in the announcement, the Board has adopted a change in the Bylaws regulating the transfer of Autodesk stock; since two directors are interested parties, we have made it subject to ratification by the shareholders. The

management's position is that we have an anomalous situation when a corporation (call it XYZ Corp) owns stock or warrants in Autodesk: our bylaws prohibit XYZ from selling its Autodesk securities to anyone else without offering first refusal to the rest of us, but there is no effective way we can keep it from selling *its own* stock to somebody we don't like. When it does so, the beneficial ownership (and voting control) of some Autodesk stock is changed in a way over which we have no control.

On the other hand, if XYZ distributes its Autodesk holdings to its own stockholders, there is no change in beneficial ownership of Autodesk and a minimal change (toward fragmentation) in voting control; and stock which is distributed to individuals will fall under the effective control of our stock transfer restrictions. Therefore, if the stock transfer restrictions are a good idea, it seems to be in our interest to encourage and expedite such distributions, and we want to remove them from the first refusal process.

The whole question of stock transfer restrictions is controversial and will surely be a topic of discussion at the meeting, quite apart from this special case.

We expect to have a distribution of employee stock options ready for consideration by the founders. This will correspond pretty much to the second and third (and last) semiannual option distributions that we planned when we organized the company 18 months ago.[162] We would like to get these options issued and out of the way before any potential investor can try to water them down as a bargaining point.

Daniel Drake
Secretary

[162]Dan Drake comments, "It should be noted for the benefit of anybody trying to do this over again that the option grants as set up by the Board were to be approved by a majority vote of the founders, one person one vote, before the Board would formally vote them. The allocation was so wildly unpopular that the chairman of the meeting, who wasn't wildly enthusiastic himself, was barely able to railroad them through, believing as he still does today that absolutely no compromise could command a solid majority. I.e. this idea didn't work very well."

Autodesk, Inc.
150 Shoreline Highway, – Bldg B
Mill Valley, Ca. 94941
(415) 331–0356

September 29, 1983

Notice Of Annual Meeting

A special shareholders' meeting of Autodesk, Inc. will be held at 2:00 PM on Sunday, October 9, 1983, at the company's main office at 150 Shoreline Highway, Suite B–23, Mill Valley, California.

The meeting was called by order of John Walker, president, by authority granted in the Bylaws.

Stockholders of record September 28, 1983, will be eligible to vote at the meeting.

The purpose of the meeting is to discuss plans for increasing the company's capitalization, possibly by attracting additional investors, and to take actions that may be required in order to acquire new investors or to make the capital structure of the company more attractive to investors. Such actions may include changes in stock option plans or in restrictions on stock transfers.

The Board of Directors will submit for ratification an amendment to the corporate Bylaws, adopted September 7, 1983, allowing a particular class of stock transfer to proceed without the offer of first refusal rights to all shareholders. The position of the management on this change is given in the accompanying letter. The management knows of no other resolutions which are proposed for adoption at the meeting. If you are not absolutely sure that you will attend, please sign and return the enclosed proxy; it will be revoked automatically if you do attend.

Daniel Drake
Secretary

This was Autodesk's first expense report form, with items befitting the proclivities of the founders. This copy was painfully reconstructed from a paper copy of the form by Duff Kurland.

Expenses

Name _____ Date _____

Address _____ Phone _____

	Date							Total
→								←
Meals								
Lodging								
Airfare								
Auto Rental								
Taxi/Limo								
Telephone								
Mileage .20/mi.								
Business Meals								
Pizza/Chinese								
Decadence								
Supplies								
Software								
Hardware								
Books/Manuals								
Miscellaneous								
Totals								

Business Meals & Entertainment

Date	Amt	Place & City	Guests	Affiliation	Purpose

NOTE: Some expenses, while legitimate, cannot be reimbursed by the company at this time. They are shown here for tax purposes and possible future reimbursement.

Subtotal	
Less Advances	
Less non-reimbursable expenses	
Balance Due Autodesk	
Balance Owed Employee	

Signature _____ Date

Approved _____ Date

Piece of Cake

If you eat beans, you fart. If you build a successful company, you hire an advertising agency. Both are laws of nature, and raging against the inevitable consequences of one's actions makes no more sense than standing on the tracks and arguing with the Twentieth Century Limited.

The wise man sees the amusement inherent in the inevitable and enjoys it to the fullest. Before COMDEX 1983 we hired an advertising agency to help us "tell the AutoCAD story". By hiring an agency one employs "creative talent" not present in the mundane people who create new technologies.

This is what they came up with. We didn't use it.

NOW YOU CAN TEACH YOUR PC TO DRAW

AutoCAD/end user ad
October 7, 1983

Just slip in your AutoCAD graphics software disc and you're ready. Draw a brick and AutoCAD will draw you a wall—automatically.

Move it. Copy it. Modify, rotate, or scale it vertically and horizontally. Store it. Change your mind and erase it.

Do it all on *your* PC. AutoCAD is the industry standard. It's compatible with most any PC. IBM PC and XT. Zenith Z100. Victor 9000. NEC APC. Columbia. Eagle PC. Not to mention CP/M-80 computers.[163]

It's just about ready[164] for NCR Decision-Mate, DMS, DEC, Sony, Televideo, Eagle 1600, Texas Instruments, and Corona.

And it supports a bunch of input and output devices.

You'll work better. And easier. Use a light pen and on-screen menus. A digitizing tablet. A keyboard or a mouse.[165]

Use them to draw lines. Of any width. Circles, arcs and solid-filled areas. Insert them anywhere in your text.

[163] But we will anyway, just to be sure.
[164] Any day now.
[165] Even a verb occasionally.

And do it in German. Or French. Swedish is coming and so is Italian. We're working on Japanese.

You won't flinch at the price. It's good. Real good.[166] $1,000. Add another $500 and you get automatic dimensioning.

Ask around. You'll find a lot of people know AutoCAD. We've already shipped more than 1,500 systems. All over the country. All over the world. And you wouldn't believe who some of our customers are.[167]

Architects love it. So do engineers. So do designers. So will you.

You wouldn't *have* a PC without a word processor. Or without a spread sheet. Or without AutoCAD.

Word processing for graphics. Take a byte. [168]

IT'S A PIECE OF CAKE.

For a demonstration and information, call or write.

AutoCAD, Inc.
150 Shoreline Highway — Building B[169]
Mill Valley, CA 94941

[166] I'm not making this up. We paid good money for this hooey.

[167] Would you believe.... Excuse me, my shoe is ringing.

[168] In addition to what we spent for this copy, we also spent $700 to get a guy to come to Greg Lutz's house and take pictures of his CGA screen with a drawing of a piece of cake on it. Of course, that looked so awful that we wouldn't have used it even if we'd run the ad.

[169] We were still concerned with looking like "a real company". I suggested that we list our address as "Autodesk, Inc., AutoCAD Division—Building B, Mail Station B-20, 150 Shoreline Highway, Mill Valley, CA 94941". Cooler heads prevailed and we didn't do it.

Information Letter 10

Here it is. John Walker recommending that Autodesk go get venture capital, even if it meant surrendering the autonomy and sense of control over our destiny that was the main reason for starting the company in the first place. But read the arguments—and remember that they were the very same arguments that possessed us to go public in 1985. This was the first "modern" information letter: written for a company in which founders were outnumbered by employees hired after the company began to expand, and after the company had significant sales and a regular operation. Thankfully, the technology crash of '83 and our unwillingness to sacrifice our principles at the altar of Mammon precluded our obtaining venture capital. No, there was no Information Letter 9; when I wrote this one, I didn't have copies of the prior letters and I miscounted.

Autodesk, Inc.
Information Letter # 10

by John Walker
Revision 8 — October 25, 1983

> In periods of frenzied haste toward wealth, of feverish speculation and of crisis, the sudden downfall of great industries and the ephemeral expansion of other branches of production, of scandalous fortunes amassed in a few years and dissipated as quickly, it becomes evident that the economic institutions which control production and exchange are far from giving society the prosperity they are supposed to guarantee; they produce precisely the opposite effect.
>
> Peter Kropotkin, 1880

Indeed. . . .

Well it has been a long time since the last information letter, hasn't it? These letters only get written when things are sufficiently calm that there is time to reflect enough to put one together, or when the gusts of crisis blow so hard that one is forced to communicate to share the urgency. Up to now. This letter is a review of our accomplishments, an attempt to summarise our current position, and a sales pitch for my recently acquired view of where we should be going.

My original working document for the formation of this company began with the words "the game has

changed".[170] The game has indeed changed, and we have changed with it. Our success to date is a measure of our ability to change and adapt to the software game in the 80's. But the game is continuing to change, and we must continue to adapt or else we will lose the opportunity we've created by our labours to date.

We started a company composed of highly skilled programmers. Our idea was to develop multiple products. We did that. We planned to test market these products and determine which had the best potential. That was done. We intended to focus our efforts on the product which, in test marketing, showed the best prospects of being a success, and we did it. We decided from the start that we would have to recruit a marketing professional to build an effective marketing team to promote our products. Since joining the company, Mike Ford has done an amazing job of expanding our sales from zero to the million dollar level, and is building a marketing organisation as competent as the original technical team. We knew that we couldn't succeed while remaining a cottage industry. When the time was right, we had to take on the burdens of office facilities, manufacturing machines, full time employees, and the rest. Since his joining the company, John Kern has made the operational side of the company professional, efficient, and responsive, while reducing the cost of our product as sold by over 50%. We are now positioned to expand our production capacity by a factor of ten without catastrophe.

So far we are on course, profitable, productive, and positioned in the fastest growing corner (CAD/CAM) of the hottest industry (Software). In the historical course of things, we should plow our profits back into the business, add to our production and marketing capacities, and continue R&D to enhance our product. But these are the go go 80's, where product lifetime may be measured in months, and company lifecycles in one to two years. As long as all our competitors play by the rules we're using we will win. But will they?

> We are all at a wonderful ball where the champagne sparkles in every glass and soft laughter falls upon the summer air. We know, by the rules, that at some moment the Black Horsemen will come shattering through the great terrace doors, wreaking vengeance and scattering the survivors. Those who leave early are saved, but the ball is so splendid no one wants to leave while there is still time, so that everyone keeps asking, "What time is it? What time is it?" but none of the clocks have hands.
>
> Adam Smith, *Supermoney*

How lucky we have been! I can think of almost no parallel in the history of the microcomputer business where one company had the only product in a major market for over a year. Had we entered a market like word processing, BASIC compilers, or spread sheets, we would have long ago been blown away by better organised and capitalised competition. It is simply fantasy for us to assume that we will continue to have this market to ourselves, or to assume that our primacy in the market guarantees success down the road. Review my initial papers about this business. Remember Electric Pencil? Has anything changed? Let's not be deceived by our initial success. We've all worked hard, and we've been lucky, but from now on in we have to make our own luck because the competition is on the way.

Our continued expansion plans can be disintegrated almost overnight by any one of the numerous competitors who are appearing on the horizon. There is little doubt that our product is technically the best, and so far we have been able to react faster and complete development goals in a more timely fashion than the competition, indicating that our technical staff is the best.

However, we are weak in our ability to promote our product. We have bootstrapped this company from the very start. This mode of operation preserves control in our hands, but it limits our ability to counter competition,

[170] Well, actually this was in the third section of the working paper (page 14). But I didn't have a copy at hand when I was writing this.

particularly in advertising blitzes, to the monthly profit from operation. For September, this number came out to a little over $13,000, which will buy from 3 to 4 full-page ads depending on where you run them (just page rates, not even figuring production cost). If you start to calculate in additional staff, overhead requirements, promotional material, etc. it becomes clear that we will have to run at a substantial loss to sustain even a modest promotional campaign. Our reserves are very healthy for a company at our state of development, but could be exhausted in less than three months by even a modest step-up in our marketing budget.

Therefore, the time has come to seek outside funding. The time is right; AI has a product recognised to be the industry leader, so we can build our having arrived first into a large market share by a properly targeted campaign. Second, this is an ideal time for a company of our kind to raise money—software is one of the hottest items now, and CAD is one of the best bets in software. This may not be true in six months. Not long ago, venture capitalists were tripping over themselves to fund start-ups in desktop computers. Today such ventures are shunned as hopeless charges against the IBMonolith. Remember gene splicing?

If we raise money now, we will get more, and give away less, than if we wait. By waiting we are gambling that our continued growth will not be aborted by a well-funded competitor, that our growth will add value to the company faster than our competitors grow, and that if we later need money we will be able to find it on attractive terms. All of those are very uncertain propositions. On the other hand, if we raise money and discover we don't need it immediately, it can serve as a cushion against hard times in the future.

Going after heavy-duty outside money will make some serious changes in this company in the process of getting it, and if we succeed, after we get it. We will have to prepare a formal business plan with sales forecasts and departmental budgets to the line-item level for the next three years. (This is a good idea anyway, and we're already doing it.)

We will have to reconcile ourselves with giving up 20 to 35 percent of the company in return for 2 to 3 million dollars in funding.

We will have to abandon any remaining ideas of running the company on an informal basis. We will be on the fast track, spending other peoples' money, and will have to show that we are being prudent and professional with the resources they are speculating on us.

We will have to accept outside participation in the management of the company. There will almost certainly be additional professional management installed, possibly replacing the entire current management of the company.

We will largely preclude operating the company as a steady-state enterprise which generates revenues adequate to make us all happy. People don't stick 3 million dollars in an untested company unless they expect to increase it by at least a factor of ten to twenty, so we'll be basically signing onto an exponential growth treadmill with serious consequences for failure to perform as advertised (e.g., loss of control of the company and serious dilution of our stock ownership).

These are serious consequences, and should only be accepted after reflection on the alternative.

> Life in the fast lane, surely makes you lose your mind.
>
> The Eagles

Ahhhh, but the slow lane has its problems too. We are now at the point in the lifecycle of our product where the market share war usually begins. Several products have announced. None has come to the attention of

more than a few percent of the potential customers. The process of new market creation is in its infancy. The entrenched large system sellers are largely unaware of the barbarians at the gate.

In fewer than ten years, drawing systems will be on the desk of virtually every person employed in a job in which drawing is a means of communication. These drawing systems will be based upon the fundamental concepts present in AutoCAD. There is little doubt that AutoCAD can do the job that is there to be done. Whether those machines run AutoCAD or another package will be largely determined in the next year. Our actions will decide whether AutoCAD becomes synonymous with "drawing tool" as VisiCalc has become with spreadsheet.

Our technical edge is only a tenuous advantage. Every day the success of the IBM PC proves that the race does not necessarily go to the best product. One must have the best *company*, and that is a composite of the product itself, the development staff which maintains and further develops the product, the customer support staff which largely determines how the product is perceived after sale, the production facility, which controls whether delivery commitments are upheld and manufacturing quality maintained, the marketing department, which must build the visibility of the product and expand the channels of its distribution, and the overall management of the company and of the departments.

But just as important as these obvious components of a successful company is its financial position. It has been proved again and again that a perfect company with the ideal product can be wiped out by a competitor who is better capitalised. For example, who would argue that the IBM PC is a better machine than (name the competitor you like best!). But IBM is not only a superbly functioning company, there's the matter of their 4.3 billion dollars in the bank. When a company like Victor raises a sum like $35 million by going public, this amasses a war chest that is insignificant against IBM's television advertising budget alone.

Not only is our current technical and market advantage tenuous, it can evaporate in the face of well funded competition. Our technical development is still very slow, very informal, and basically built on a few key people burning themselves out to get work done. I know all about large software projects and the false economies of scale, but the fact remains that if, say, Digital Research plans to enter the CAD market, they can field a product within 9 months which would be better than AutoCAD by most technical measures.[171]

Next, there's the fact that simply by obtaining outside funding we add an aura of credibility to the company and make ourselves more visible. I don't like the fact that companies who bootstrap themselves are sneered at, but that's the way it is, folks. And with a 46% tax on profits before you pour them back into development, it's a lot easier to do it with other peoples' money than with your own. By being in the portfolio of a well known high technology venture capitalist, we will be exposed to possible joint venture opportunities or markets for our products which we would otherwise have to painfully find for ourselves.

Then there's the experience we can draw on in addition to the money. Before we started this company, none of us had started from zero and built a company to the million dollar scale. I'm sure that had we done it before, we would have done many things differently, and made fewer painful mistakes. I know I would do it differently if I had it to do over. Now we have to face the fact that none of us has built a million dollar company into a 30 million dollar company. This is a lot faster, more high stakes game than the one we have been playing so far, and we will need a lot more than honest intentions and a couple of books to survive in it.

The point is that these venture capitalists have done the trick numerous times. They know what you have to do when, what to do when you get in trouble, and where to find the people you need when you have to build up a critical part of the company. In growing and keeping out of trouble, this knowledge can contribute mightily

[171] This was a 1983 version of the "Nightmare Scenario" I spoke of at greater length in 1991's *Information Letter 14*. See page 625.

to the chances of success. And with 80% of new businesses failing within 5 years of formation, we can use all the help we can get.

> It's better to burn out, than it is to rust.
>
> Neil Young

So, all things considered, it seems to me that the time has come for us to seek outside financing. We should attempt to raise from 2 to 3 million dollars by selling up to 35% of the company. We will almost certainly have to accept serious downside risk in the form of loss of control or dilution should we fail to meet performance criteria spelled out in the financing agreement, so we will be betting everything on success and risking all we have done so far.

But it really boils down to this: take your present percent ownership of the company. Would you rather have that percentage of a company with a market value of $3 million, or 65% of that percentage of a company with a value ten to twenty times as great? I think that the risk of failure is very great, but that we can fail just as badly by not taking the risk—since we are not operating in a vacuum and even our present market share may erode to the point of disaster against a well financed group of competitors.

I have come to this recommendation from a position almost completely opposed to what I'm recommending now. I have changed my mind because I think that we have in our hands a product which can make us all very well off, indeed. As much as I dislike it, I believe that the only way we can get from where we are to where we want to be is to bring in outside money and talent, and rapidly make this a professional, conventional company.[172] Not to do so is to subject our work to date to a greater risk of loss due to competition.

The next year or two are going to be very interesting ones. We are going to be jumping on a tiger's back and trying to hold on as we learn how to steer. But I can't think of a better time, or a better industry, or a better product, or a better team of people with which to make this gamble. And I can't imagine starting over if we let this opportunity slip away.

> But with the throttle screwed on there is only the barest margin, no room at all for mistakes. It has to be done right... and that's where the strange music starts, when you stretch your luck so far that fear becomes exhilaration and vibrates along your arms.
>
> ...letting off now, but only until the next dark stretch and another few seconds on the edge... The Edge.... There is no honest way to explain it because the only people who really know where it is are the ones who have gone over. The others—the living—are those who pushed their control as far as they felt they could handle it, and then pulled back, or slowed down, or did whatever they had to when it came time to choose between Now and Later.
>
> But the edge is still Out there.
>
> Hunter S. Thompson, *Hell's Angels*

[172] This was, in my mind, the point at which the original rather odd concept and culture of Autodesk ended, and the transition toward building a professionally managed business began. Even though we never actually obtained venture capital (we rejected an offer of funding in May 1984 as too little money to justify the constraints on our freedom of action—see page 233), Autodesk rapidly began to put its business house in order. Claims that, years later, Autodesk was a "cabal of hackers" or some such are nonsense rooted in ignorance.

And we're a long way from it.... Right now we are ticking along in first gear, running smoothly, and taking every precaution. But what's that noise from behind? Shall we reach for the throttle, try to hide, or hope they will just go away?

I say let's go for it.

Information Letter 11

I was far from happy with this Information Letter when I wrote it, and I'm far from happy with it as I reread it today. More than a year had elapsed since I wrote the last Information Letter. When I wrote IL 10, the key people in the company met every Wednesday night in the room overlooking the Moment's Pause hot tub, and discussed matters ranging from whether we should sell our souls to the venture capitalists to ink running on the metal labels. When it came time to write IL 11, there was this real company: multiple buildings full of people at 150 Shoreline, different departments pulling in different directions, and all the inevitable concomitants of a growing, vital company made up of incorrigible individualists.

When I penned this letter, I knew that we had made plans to invite Hunter S. Thompson to our party. I pulled that punch because I was afraid it would fall through (it didn't—he showed up), and because I was worried that our grand scheme to have him chronicle the evolution of Autodesk over the next several years, thus creating the Marin County Gonzo rejoinder to wimpy Boston's Soul of a New Machine*, would fall through (it did—and in my opinion, Dr. Gonzo blew an opportunity to be for High Technology what Hemingway was for his generation).*

This was the only Information Letter I deliberately wrote thinking about the audience. Think of it as having been written on that cusp between being a company of close friends and a major force in the market, or simply the time that John Walker chickened out.

Autodesk, Inc.
Information Letter # 11

by John Walker
Revision 4 — February 12, 1984

Party Time

On January 16, 1984, Autodesk's sales for the fiscal year which began on February 1, 1983, went over a million dollars.

"This situation absolutely calls for a really futile and stupid gesture to be made on somebody's part. And we're just the guys to do it."

On Sunday, February 26, 1984, we'll have a party to celebrate this accomplishment. We don't know yet exactly where the party will be held, but it will start in the late afternoon. Plan to be there.

On the scale of the universe, selling a million dollars in a year may not count for much, but on the scale we've been used to it is a major milestone, and provides a good excuse to think about where we've been and where we want and hope to go.

What follows is probably the most disconnected and rambling Information Letter you have received. This IL is not written at a major event in the history of our company, or at a time of serious crisis. There's not a consistent thread to connect the thoughts which follow. This is, however, a time of rapid growth for our company and a time of rapid change in our style of operation. This is, in itself, cause to look at what's really happening.

And you see, our rapid evolution is really the subject of this letter. Our success contains both the seeds of our future success and the potential for our undoing, because as we rapidly expand the company as we must, we unavoidably change the character of the company, and risk destroying the things that have made us a success. I know that in my round the clock bursts of effort, I have not made the time to talk to everybody as much as I should have, so I'm taking this opportunity to let fly with a collection of random thoughts and questions. This is not supposed to replace conversations, just start them going. I am interested in talking with everybody associated with this company about what is going on and how we can best align the company with your goals, so consider this an invitation to corner me and start talking.

What's Going On Here

If things associated with this company are chaotic, there is a good reason. Starting in October, our sales volume has been growing at an average rate of over 30% a month. As this has happened, the workload on everybody has increased at this exponential rate. The telephone volume, manufacturing, shipping, and customer support loads have tracked these increases. Although we have been adding new people to bear the load, and will be taking additional space to avoid packing people like sardines into the current building, it takes time to find good people and time for them to learn the job, time to locate office space and move into it, and time to remedy the execrable phone "system" we have now.[173]

For the moment, about all we can do is hang on and wait for the solutions to these problems which are on the way. Even in the most foamy bubbles of optimism, nobody expects a 30% compounded month to month growth to continue. If it did, our monthly sales at the end of this year would be over 2.8 million dollars *per month* (an annual rate of 33 million per year), and at the end of next year would be running at the rate of about half a billion dollars a year. Hi ho. Look out General Motors.

But seriously folks, what we can most reasonably expect is a series of plateaus with up-slopes between them. We're on one of those giddy up slopes now, but I think we'll have a chance to recover before the next expansion. But then, let's not get too cocky....

New Entrants, Undeterred by Mounting Casualties, Crowd the Software Field.

[173] You see, we were in this *motel*, and everybody had their own phone. If you wanted to hand a call off to somebody else, you had to ask them to call that person's number and hope they'd be there to answer.

> Sofsearch International, a San Antonio, Texas company that helps computer users locate software has lost track of almost 1700 vendors in the past year.
>
> No matter, the company's count of active software vendors is nonetheless up about 57% from a year ago, to 13,500. Says a distributor, "For every supplier that goes away, another 15 or 20 come up behind".
>
> But prospects may be poor for many of the new suppliers, "Before word of mouth was enough to make your product known.", says a distributor of software, "Now it takes national advertising and huge budgets."
>
> *The Wall Street Journal*, February 8, 1984

We are indeed playing in a very different arena than the one we entered when we started this company. We started out with some product ideas, a commitment to work hard for little or no money, and the idea of building a company which would develop creative products, provide the highest standards of quality, do well by our customers in both service and responsiveness to requests, and reward those who did the work with the fruits of their labours if and when the company was a success.

Platitudes maybe, but all of us have worked lots of other places that didn't even pretend to do things that way.

And what do you know, it's working.

Now we have to face the challenges posed by our successes so far. We have to build the company rapidly, maintain the safety factor which has allowed us to survive lean times in the past, and continue to adhere to the principles that have been working so well for us up to now. This may be a lot easier to say than to do. There is a powerful force which pushes organisations toward mediocrity and insensitivity as they grow, and resisting it must have a high priority here.

And yet, we must change. We are working with much larger sums of money, many more people, numerous outside consultants and vendors, and with a vastly increased workload. The old informal channels of communication just cannot handle the load any more. We have been and will continue to institute the internal procedures we need to make information flow smoothly and to enable us to make the most of our limited funds. We must control our money very carefully now. Prudent use of our money now is one of the keys to making this company a success, so we will continue to watch expenses very closely.

If we obtain an infusion of money from venture capital sources, this will only increase the need to manage our money very carefully. When we obtain outside funding, we are committing to use that money as carefully as we can to make the company a success. We will control and deploy that money even more carefully (if that be possible) than the hard earned dollars we invested in this company or the sweat-stained dollars we pour back in from the sales we've made so far.

> **Explosive Growth Foreseen In CAD/CAE**
>
> The computer-aided design (CAD) market will reach $6.9 billion by 1987, according to the Yankee Group (Boston). CAD industry revenues were $1.3 billion in 1982; the CAD market is expected to grow at 40% annually through 1986.
>
> *Systems and Software*, February 1984

Who can doubt that we are in the right place at the right time?

Now let's talk about what isn't supposed to change as this company grows.

This company was formed as a vehicle through which people who had in the past worked very hard with little reward could not only reap rewards, but control their own destinies and make the decisions, for better or worse, which would determine whether we succeeded or failed. This company has never put people into boxes or told them that they couldn't try something they wanted to do. There has always been more work to do than people and hours to do it, and nobody who has asked to try something has been refused a chance at it.

> The computer programmer, is a creator of universes for which he alone is the lawgiver. Universes of virtually unlimited complexity can be created in the form of computer programs. Moreover, and this is a crucial point, systems so formulated and elaborated act out their programmed scripts. They compliantly obey their laws and vividly exhibit their obedient behaviour. No playwright, no stage director, no emperor, however powerful, has ever exercised such absolute authority to arrange a stage or field of battle and to command such unswervingly dutiful actors or troops.
>
> Joseph Weizenbaum, *Computer Power and Human Reason*

But in the real world it's a whole lot harder. Our freedom to make decisions is constrained by money, by time, by the realities of working in the real world with real people. By succeeding in this domain as well as by creating the best computer program of its kind, we can gain the real rewards, both material and internal, that we started this company to achieve. But is it any wonder that it's easier to program up a storm than come to terms with this rapidly growing company? Between the risks inherent in the software business (where else can somebody copy a $1500 product for $5, or destroy an industry by a Kamikaze marketing strategy of, say, $50 per?) and the potential of being the leader in the fastest growing corner of a business growing 40% per year, is the reality we make out of the opportunity we have created by all those nights we worked straight through.

And that reality can be best served by making this company work as well on a large scale as it did when it was just a few wild eyed maniacs in various basements. This means that we have to make the company responsive to what people want. This is a lot harder to do as we necessarily increase the distance between the people developing the product and the people using it in the field. But we have to do it. We need to feed ideas back and forth rapidly. Anybody may come up with an idea which could make AutoCAD usable to a whole new group of people. Anybody may hear a customer suggest such an idea. We have to make sure that ideas like these aren't forgotten or left unacted on.

This product and this company aren't successful because we've spent loads of money advertising and whipping up a demand for a product. We've done so well because we created a product which fills a basic need. This is a product which excites people by its very existence. It's fun to use, and it lets people do work they couldn't otherwise do without spending hours of tedious labour. This product has put in the hands of the individual and small company the power which previously was only available to large companies—which contributes to leveling the playing field and eliminating advantages of scale.

We can continue to build on this success without losing track of how we got here. We have to continue to listen to each other, to customers, to anybody with a bright idea. We can't let schedules, budgets, meetings, departments, and memos dim the spark of creativity which built this company, or structure out the immediate communication of ideas and rapid response to requests which firmly established our reputation as "good guys".

When we advertise, all we're trying to do is tell the 98% of people who can use our product, but don't know that anything like it exists, "Hey, look here". When we try to obtain publicity, we can succeed best by telling stories of people who solved problems by using AutoCAD. These ideas and these stories will not be dreamed

up by advertising people—their job is to communicate, not to invent. We have to find these stories and present them in a form where they will be understood.

And so we must not look at advertising as a black box where you feed money in one end and sales come out the other. It's one of the many means of communicating. More important than the advertising budget is what we want to communicate. Who are we? What do we sell? Why do people need it? What problems does it solve? Why is ours better than theirs? We have to answer these questions. Nobody else can.

And so, I see our challenge here as mainly keeping on track. If we can continue to be responsive, to act quickly, to get a lot done with a little money, to make one piece of work benefit us in multiple ways; if we can continue to make this company a humane place to work where people are rewarded for their intense labours and where anybody can advance rapidly just by carving out additional responsibilities, then I think that the success we have experienced so far will be multiplied by many times over the next few years.

The Slingshot and Success

I'd like to wind up with one of the most obscure metaphors I've ever used in one of these letters. In celestial mechanics there's a concept known as gravity assist, or the cosmic slingshot. It's how the Voyager probe got to Saturn.

If you take a satellite and drop it down very close to a planet, then fire its engine, the power of the engine is multiplied by the gravity of the object you're whizzing past. Any effort made at that peak point, at the ragged edge of plunging into the pit, is multiplied thousandsfold versus efforts made in the calm void, far removed from risk and turbulence.

We're in the heart of the maelstrom now. We're growing so fast we can hardly keep up. We're becoming known, and how we treat people and how well we meet their needs now will determine how we're perceived for years to come. The reputation of our product is being made on a day to day basis. One major screwup and we can lose it all—overnight.

What you do now in this crazy environment will cast a long shadow on the future of this company, on your career, and on all of our hopes to share the rewards of success.

This is not the time to coast. This is not the time to let growth squeeze out innovation or our ability to take a risk or grab for the main chance. Today, this company looks awfully goddam respectable compared to where we were, say, last May. The trappings of success are everywhere. But remember, if we get to where we hoped to be when we started this venture back in January of 1982, we'll look back on these as the "old days". To coast now will make this the peak, not the stepping stone to where we want to be.

Every dollar we spend, whether for salaries, rent, raw materials, or advertising, was generated because somebody chose to buy our product. That person looked at the fruit of our labours, looked at our company and our commitment to help him after he bought the product, at our future and the promise that held for future development and additions to improve what he was buying, and decided that what we had to sell, which is nothing more than a great idea written onto a $5 floppy disc, with a manual, was worth more than $1000 or $1500 in his pocket.

Whether we have the resources to continue our growth, whether we survive or join the ranks of the software companies that "drop from sight" depends on whether people continue to value what we have to sell as worth

more than that money in their pockets. Buying anything, but especially something as intangible as a computer program, involves putting your trust in the person who's selling it.

If we continue to deserve that trust, we'll do very well, indeed.

The Deal on the Table

After talking to virtually every venture capitalist in the business, in May of 1984 it appeared that we were finally going to close a deal. Frank Chambers, who had been introduced to us by Jack Stuppin, indicated that he was willing to make an investment in our company. Despite our cynicism, born of endless tiresome and fruitless meetings which consumed our time when it was desperately needed to develop the company, we were eager to obtain funding we could apply to increase our marketing efforts and seize the market before better-funded competitors entered the fray.

Mike Ford described the situation as "being in a verdant field with gold bars lying all around. The question is how many can we throw in our pickup truck before a big vacuum cleaner comes down from the sky and sucks them all up." In essence then, being greedy and aggressive suckers ourselves, we wanted the cash to nail down the largest possible market share while we still could.

After numerous and lengthy discussions, Frank Chambers communicated the terms of the deal. What follows is a transcription of Dan Drake's notes taken while hearing the terms over the phone, with annotations explaining what the terms mean.

Terms I: Frank Chambers

- $500–700K. Size of the investment in dollars. This would represent about 1/10 of the company.
- $2 preferred, convertible to 1 share common. The investment would be preferred stock, so that if the company failed the investor would get his money from the remains before any of the common shareholders. The stock could be converted to common stock at any time. This is conventional in venture capital deals.
- If not liquid in $4\frac{1}{2}$ years, we offer to repurchase at $2\times$ price (plus accrued dividends). If the company did not go public or get acquired, thus providing an opportunity for the investor to "cash out", we would buy back the investment at twice its original value.
- 8% dividend from 2/1/85; with majority of preferred holders or the board able to substitute change in conversion ratio. We would pay this dividend from the company's earnings. If we didn't have the earnings to pay the dividend, the foregone income would be used to increase the venture capitalist's ownership share of the company. Most venture capital deals work this way, but it's unusual to have the dividend start so soon after the deal.
- One demand registration, unlimited piggyback. The investor could, at will, demand that we make a public offering in order to enable him to cash out his stock. The company would bear all of the

costs of the public offering. In this, or any other offering the company made, the investor would be able to sell his shares "piggyback", up to the total amount of the shares owned.

- Options.
  - 110,000 at $.75.
  - Remainder at least $2 + 5% per 6 months.
  - Dilution protection, unspecified, above 500,000 options.
  - Vesting at least 4 years.
  - Forfeited options are canceled, not returned to pool.
- Board of directors. 7 people: preferred elects one, one by agreement of preferred and common shareholders. Jack Stuppin to be on the board. Advisory committee to the board up to 3 people chosen by the investor.
- Representative of preferred investor approves: "All compensation matters", and all capital expenditures over $25,000.
- John Walker: key man insurance $500K; employment contract, 2 years, non-competition.
- Frank Chambers approves investors.
- Preferred has first refusal on private equity offerings.

We didn't like it. While many of the terms were conventional and were what we expected, several totally unexpected constraints on our ability to develop the company in the way that had brought us to the present point were contained in the deal. In particular, our ability to grant stock options to new employees was severely constrained by limits on the number available, by forcing the option exercise price to above the price paid by the investor (who received much better terms on his preferred than the employee would on his common stock), by retiring from the pool any options granted to an employee who subsequently left the company, and by imposing a four-year vesting period on all options, which the founders of the company felt transformed the options from their original purpose of allowing employees to share in the company's success to a kind of twentieth century indentured servitude which compelled employees to stay with the company or face forfeiture of their financial gains.

We also thought that the general tone of the deal was far from consonant with the percentage of the company being purchased and the demonstrated performance of the company to date and the track record of its managers. But we still wanted the cash. So... we came back with the following suggested terms.

Terms II: Us

- $500–700K.
- $2 preferred.
- Repurchase at $1.5\times$ if not liquid in $4\frac{1}{2}$ years; call on preferred at $4 + 20% per year. The call provision would allow us to forcibly buy out the investor for twice the investment plus an increment of 20 percent per year. This was meant to be symmetrical with the repurchase provision benefiting the investor.
- 8% dividend from 2/1/85; reduction in conversion price on missed dividend. This is equivalent to the original requested terms, just more lucidly put.
- One demand registration if proceeds exceed $\$X, X \geq \$5 \times 10^6$?
- Options: dilution protection by issuing proportional shares at the same price as *exercised* options. Vesting set by board when option issued. No cancellation of forfeits—return to pool.
- Board of directors. 7 people: preferred elects one, common elects 6.
- Ceiling on executive salaries and bonuses until some numbers (sales and profit) achieved; override by preferred representative. Preferred representative approves capital expenditures over $100,000.
- John Walker: key man insurance; contract, no non-competition. Being an ornery S.O.B., I said that signing a non-competition agreement with a company in which I was the largest shareholder was beneath my dignity and that I wouldn't do it. Moral: don't pick an asshole to be president of your company.
- Frank Chambers approves investors, but present stockholders can buy in subject to $ ceiling and regulatory problems. We didn't want the venture capitalist to be able to veto further investments by people who got in before he did.

- In general, the constraints have drop-dead provisions. This meant that every constraint on our freedom to run the company would expire when we reached some well-defined performance milestones.

He didn't like it. So, we got together and attempted to come up with another offer which would be acceptable. Here it is.

Terms III: 5/10/84

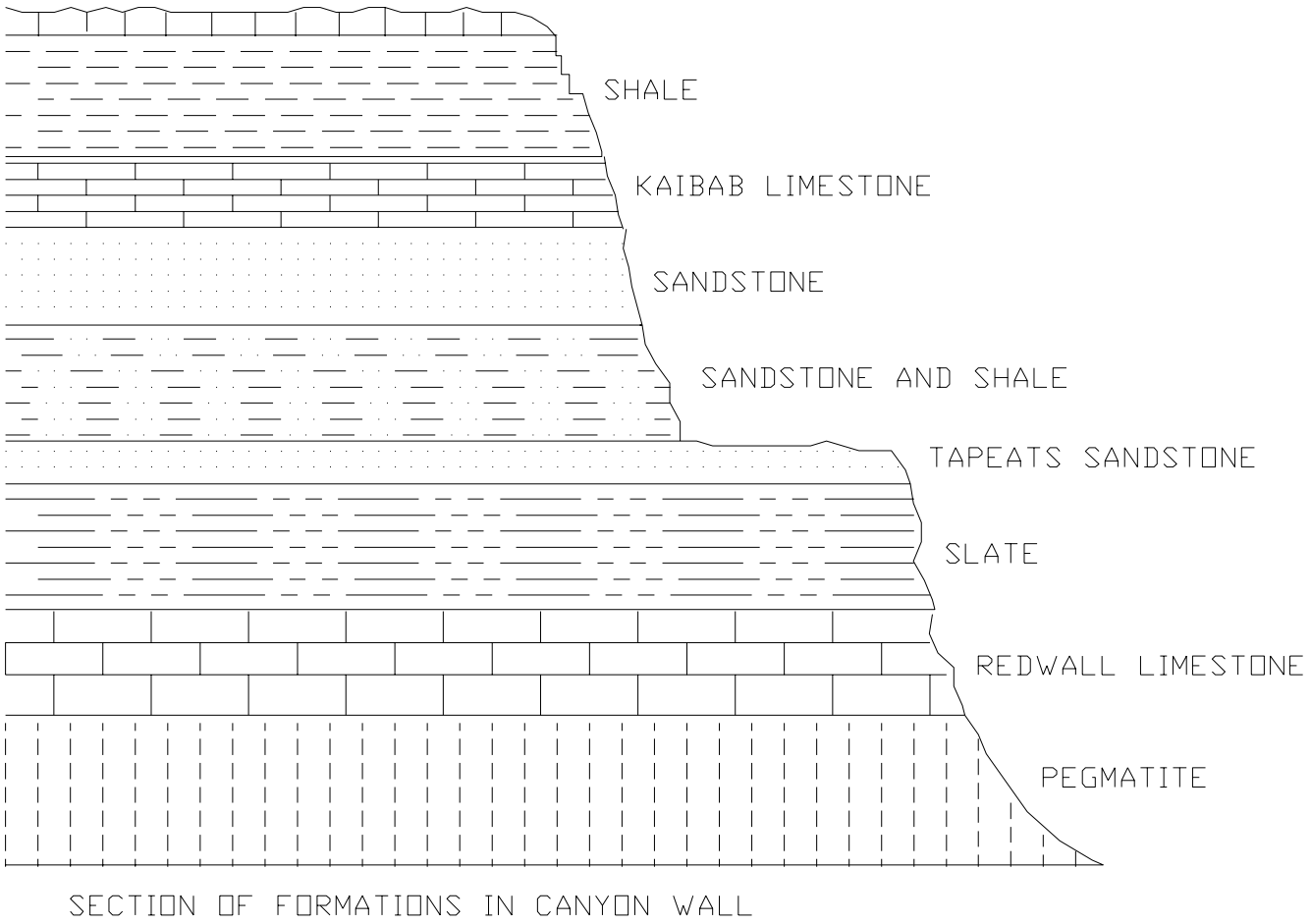
- $500–700K.
- $2 preferred, convertible.
- Repurchase terms OK (?) We acceded to the original terms.
- Forced conversion if we get to $10 million (?) annual sales and $1 million annual profit or make a public stock offering over $5 million.
- One demand registration, piggyback, *after* 1/1/87 for >$5 million.
- Options: 100,000 shares at $.75, 200,000 shares at $1.60, 200,000 shares at $2.00, 200,000 shares at $3.00, 200,000 shares at $4.00. Majority of each class of stock to approve any new plan. Vesting at best 50–25–25 after the first 100,000 shares. This meant that after the first 100,000 shares of options (which were committed to fulfilling options we'd already granted to people), those who received options would receive 50% of their shares the first year, 25% the second year, and 25% the third year. Forfeited options return to pool.
- Board: ask Chambers†, Ellison, Stuppin. (? Unsettled). What are the terms on paper?
  † If not on board, an advisor.
- $75,000 ceiling on officer and director salaries. Override by 6 or 7 directors.
- $25,000 on capital expenditures. Override how?
- John Walker: key man insurance; employment contract to "devote substantially all his time".
- Frank Chambers approves investors. Common holders can buy in (n.b. might have to keep it intrastate to avoid sophisticated investor problems). Since people who joined the company in the beginning and worked themselves to exhaustion to build it to the point that venture capitalists were interested in investing were not *ipso facto* "qualified" to risk their savings by investing in their own company since they were not already wealthy. This is an example of what was referred to in the 1980's as the "opportunity society".
- First refusal on private equity offerings.

After these terms were presented, it was clear that we would never come to an agreement on the issue of awarding stock options to employees to give them a real stake in the company's success. In addition, the overall flavour of the deal seemed to us totally inappropriate for a company which was, at the time of these negotiations, generating sales equal to the size of the deal every month and generating after-tax profits close to the size of the deal every quarter.

We couldn't believe that this was the best deal obtainable for venture funding of the company, and we were inclined to ask around to see if this was reasonable. But, our Distinguished Financial Advisor informed us that this would constitute "shopping the deal" when "a deal was on the table" which was right out by the genteel standards of the venture community, and that he could not countenance such unrefined behaviour (notwithstanding the fact that in the real world this kind of collusion is called "conspiracy in restraint of trade" and people go to jail for it).

So, after a brief weekend meeting in which we discovered we all agreed on the obvious conclusion, we decided to graciously decline this generous offer of funding and carry on with our own resources. Upon hearing this decision, Jack Stuppin said that if we didn't take this deal, he did not wish to be a shareholder and wanted us to buy him out. Not wishing to deplete our treasury, we declined. In not accepting our terms, which differed from his original proposal primarily in issues of philosophy, not money, Frank Chambers chose to forgo an investment of $500,000 which, if held until the stock price hit its 1987 high, would have appreciated to more than $37 million.

Peter Barnett drew this geological strata illustration in 1984 to demonstrate the multiple dot and dash line types introduced in AutoCAD 2.0. The drawing doesn't represent any real geological formation—it was made up out of thin air, not hard rock. It has been used as a sample drawing from AutoCAD 2.0 to date.



AutoCAD Lite

With the burgeoning success of AutoCAD, I was increasingly worried about new entrants in the market turning our own strategy against us—entering the market with a low priced package and taking the entry-level user away from us, just as we had done with the big CAD companies. The growing success of the low-priced software market, pioneered by Borland's Turbo Pascal, only intensified this concern.

This proposal, for "AutoCAD Lite", was the first real expression of the need for an entry level drawing product in our product line. Of course, we never did implement AutoCAD Lite, but it's interesting to compare its specifications with those of AutoSketch, which we announced to the world almost precisely two years later. In November 1993, almost a decade after this memo was written, Autodesk announced AutoCAD LT, a $495 subset of the then-current AutoCAD Release 12 (see page 832).

AutoCAD Lite

Suggestion by John Walker
July 7, 1984

Almost since we made our first sale of AutoCAD, we've been periodically kicking around the idea of selling a stripped version of AutoCAD at an entry-level price. These plans have always foundered as we failed to identify a usable subset of AutoCAD which could be sold in, say, the $250 to $500 range.[174]

I believe that such a product exists, can be used for serious work, is not likely to hurt the sales of our main AutoCAD product, and can be added to our product line with minimal development effort. The product is AutoCAD-80.

Over 100 copies of AutoCAD-80 have been sold to date, and many are being used by such people as Eric Clough, who is making better use of this limited product than most people make of the full AutoCAD-86. Thus I think that there is no question that the package is useful.

Conversely, reviewing the thin AutoCAD-80 manual, and remembering that having been cut out of the development mainstream at release 1.2, AutoCAD-80 lacks many features of even the non-ADE AutoCAD-86 that a professional is likely to require. Thus, if we assume that most of our current users are those serious professionals we talk about, few of them would spend all day working around the limitations of AutoCAD-80 to save 1750 bucks.

[174] Of course, by the time we started on AutoSketch, which was targeted below $100, we had already reduced the price on basic AutoCAD to $300, positioning it in this price bracket.

I am *not* proposing that we work with the PL/I version of the program! Instead, I am suggesting that we take the exact feature set of that program as the definition of our AutoCAD Lite product and make the C version down-configurable for that version. I would propose making some of the following strategic changes in the definition of the product. Features unavailable in AutoCAD-80 are not mentioned in the following table.

- Make the Lite version write its database out in single-precision.[175] Greg's SCATTER/GATHER code should make this easy to do. It would halve the database size (making the package more appropriate for floppy systems), and would restrict accuracy to 8 digits or so (the same as AutoCAD-80, which is a single precision floating point package).
- No dimensioning would be available.[176]
- Disable 8087 support.[177]
- Remove DXF input and output capability.[178]
- Remove LAYER capability.[179]
- Remove LTYPE capability.
- Remove TABLET mode.[180]
- Disable shape compiler—allow only fixed fonts.
- Develop a conversion capability to allow regular AutoCAD to read Lite drawings. Full AutoCAD drawings could not be down-converted to the Lite version.
- (I'm not sure about this) Remove pen plotter support and support only dot matrix printers.[181]

Many of the above product definition changes are intended to turn off features which require a lot of customer support time as much as to limit the usability of the package.

The resulting product would be far more usable for the kinds of things people attempt to do with other entry-level programs like Draft-Aide, Robocom, Caddraft, and Cascade I, but should not present serious competition in our present market. Providing an upward migration path to full AutoCAD would lock in the entry level user who purchased the Lite package. (We should, however, think seriously about the possibility that some of the manufacturers and/or Summit might buy this package to bundle as AutoCAD graphics rather than the full package. Would this be bad in the medium run, if purchasers upgraded at dealer price or list? I think some exploratory talks are needed if we decide to do this.)

Strategically, we should separate the Lite version from its big brother. We want the Lite purchaser to buy because of the association with AutoCAD, but our current customer not to seriously consider the Lite package for his purposes. Might the Lite package be well suited to schools where we have to cut very low-price deals? This might be a way to keep our price up on the main package but essentially give away unlimited Lites to cut

[175] AutoSketch followed this recommendation.
[176] AutoSketch not only had dimensioning, it had associative dimensioning six months before AutoCAD.
[177] AutoSketch not only supported the 8087, it had in-line 8087 support a year before AutoCAD.
[178] AutoSketch had DXF out but not in.
[179] AutoSketch has 10 layers.
[180] AutoCAD LT deleted tablet support.
[181] AutoCAD LT supported only Windows printer drivers. Most pen plotters were not supported through this interface.

the deal. Secondly, we might let Lite out into mass distribution and discounting. We wouldn't make much on each one, but the numbers would add up, and every Lite is a foot in the door for a full AutoCAD, and a sale we should be able to make if we follow up the Lite leads. Lite might be the foot in the door with people like Koala who don't understand the concept of $1500 software.

We have heard that the entry level packages have not been selling well, but of course one has to remember that most of them are terrible. We know that AutoCAD-80 isn't, because its users seem to be happy and productive (but wish, of course, they had all those neat ADE features...). My idea on the Lite would be to bring it out on a subset of machines (say, the ones Lotus supports), and immediately look for distribution through Softsel, Lifeboat, Koala, Computerland HQ, etc. We would package it cheaply, provide a limited number of machines, and plan no development at all after introduction so ongoing manufacturing and support costs could be low. We would not add hardware protection, but might use something like ProLok.

Development of such a product would be a moderate amount of work, as it would involve putting a lot of tests all over the product, removing a lot of code by conditional compilation and linking to make the thing small, and of course testing the final product. But once a version was certified fully debugged on a given machine the development effort would be zero. Porting to new machines would involve simply remaking the package with our regular driver for that machine. We could create the manual by editing the existing manual.

I am confident that a useful product can be created according to these guidelines. What we need to establish is:

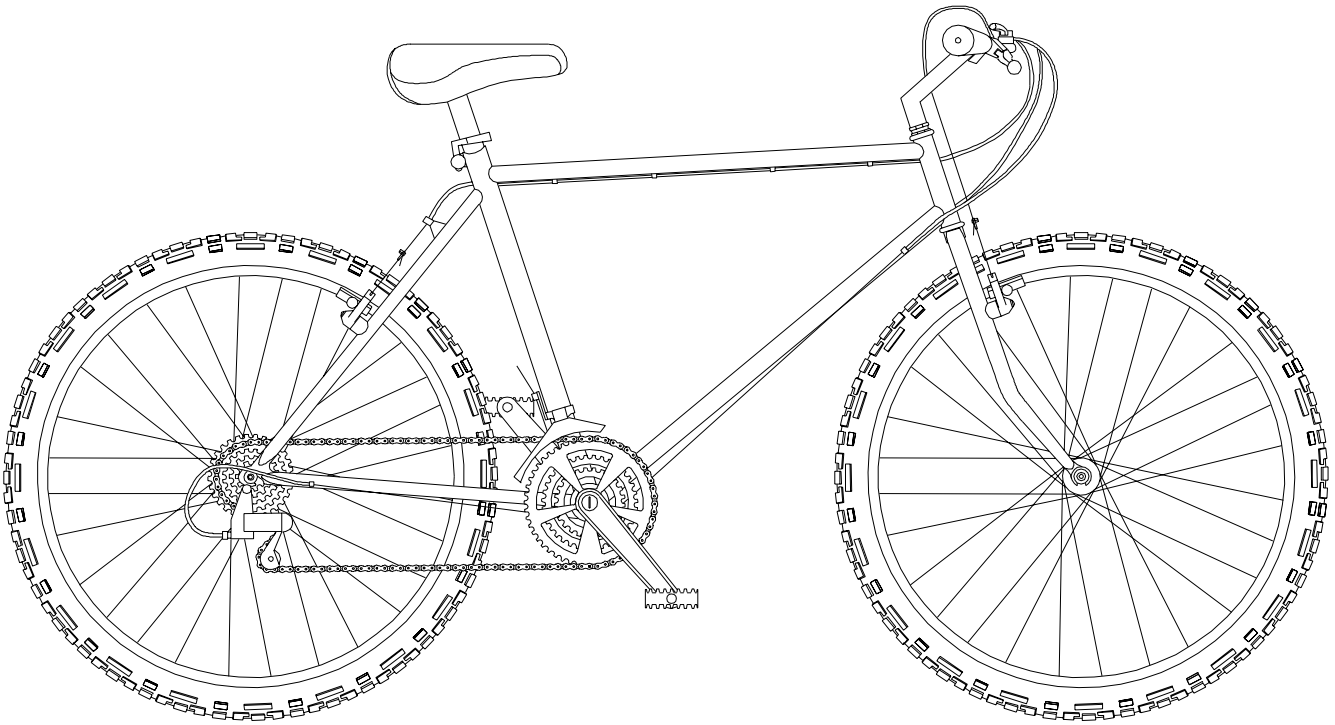
Will it hurt our existing AutoCAD sales, and if so how much?
How much support will the package take?
Where should such a product be priced?
How much can we expect to make on each one?
What are the best distribution channels?
How much will it cost us to find out if this is a dumb idea?
Do we have the time and money to launch such a product?

Finally, there's the Japanese invasion of the low-end computer market with Z-80 machines running MSX (Microsoft's Z-80 clone of CP/M).[182] If we entered this market with AutoCAD-80, we would then have a compatible 8086 product to fill out our product line.

So in one sentence, I think we can make such a product. Do we want to?

[182] An invasion which had essentially no impact outside Japan.

AutoSketch has more than fulfilled the goals set forth in the original "AutoCAD Lite" proposal. As with AutoCAD, AutoSketch users immediately began to apply it in creative ways we never anticipated, and created remarkable drawings with it. This mountain bike was drawn with AutoSketch by Larry Dea.



Taxes and Such

As the company continued to succeed, the stock price had to be revalued even though the stock was not publicly traded. This began to have tax consequences for the founders (in particular emptying Dan Drake's and my bank accounts because of a little Sin of Omission on the part of our Distinguished Legal and Accounting Advisors). Clearly, if the success continued, we needed to learn well and learn fast the rules under which we'd be playing in the stock arena.

Dan Drake researched this area in depth and wrote this memo which was originally circulated in August of 1984 and was revised and updated twice in 1985 as the public offering loomed closer. Though dated in some particulars of the law and tax rules, it's still the clearest statement I've seen of the twisty and treacherous passages one must negotiate to survive creating a new business and hundreds of jobs and not be either reduced to poverty or going to jail. Most companies don't warn their employees about any of this; the investment bankers were amused that Autodesk "took the risk" in giving this advice to the people who had built the company to the point the bankers could take it public.[183]

To: All stockholders of Autodesk, Inc.
From: Dan Drake
Subject: Taxes and such
Date: August 26, 1984 / March 6, 1985 / May 8, 1985

> "I have yet to see any problem, however complicated, which, when you looked at it the right way, did not become still more complicated."
>
> Poul Anderson

[This is a re-issue of a piece we distributed last summer. Not much has been changed, except to correct a couple of errors; in particular don't look for realistic prices in the examples. One of the errors, by the way, was based on published information from a Big Eight accounting firm. Does that mean that even they don't understand the law? Impossible.]

Some people have asked me for a summary of the stuff that I've learned about the landmines that the IRS, SEC, et al. have strewn in front of us. This is it. You should, as the saying goes, check with your own legal and financial advisers before believing any of this.

[183] This document should be read as a mid-1980's period piece, *not* a description of current tax law. Much has changed since then.

Much of this discussion concerns what happens when you sell stock, but don't get too excited: at the moment it's nearly impossible to sell our stock without going to jail. Sometime, though, our stock or a successor stock will be registered with the SEC so that it can be traded like any other, and you'll have to know the rules. If you persist to the end of this thing, you'll find some of the really amusing rules that apply when the stock is public, including Rule 144 and The Amazing Sixteen (b).

Actually, it may not be too hard to sell stock if the buyer is a California resident who already owns our stock, or if no citizens or residents of the United States are involved. (If you have reason to do it, ask for the details). Even so, it would be troublesome to trade the stock actively, but you've already signed an investment letter asserting that you have no intention of doing so before the stock is registered.

Warrants

(The first two paragraphs are obsolete, but the rest may be of interest to people who have exercised warrants.)

When we first sold stock, we issued warrants as a sweetener to encourage people to invest hard cash. These allow you to buy additional stock at $.10 a share, provided you do it by April 30, 1986. To exercise (buy the stock) you fill out the form that's attached to the warrant, write a check, and give both to the corporate Secretary (me). The company will issue a stock certificate as soon as possible.

You can, in principle, sell warrants to other people; but the restrictions are as bad as those on selling stock.

The only tax problem associated with warrants, as far as I can tell, is that you want to watch the capital gains rules. From now until 1989 (?) this means that you want to exercise the warrant at least six months before you sell the stock; then you pay income tax on 40% of the difference between the selling price and the $.10 that you paid for the stock. (Does the $.001 that you paid for the warrant come into this? I think so, but it doesn't make much difference.) If you sell for $5.00, this means taxable income of $\$0.40 \times (5.00 - 0.10) = \$1.96$; this tax will not exceed 50% of the taxable amount, or 20% of the total gain, or $.98, unless they change the rules again.

If your tax bracket is below 50%, you will pay even less than 20% on your capital gains—but it's just possible that you'll get caught by Alternative Minimum Tax (see below) and have to pay 20% anyway.

While you're waiting for the capital gains treatment to ripen on the stock you got for a warrant, you're free to sell stock that you bought before; just be very careful to turn in the right stock certificate and to keep proper records. (Well, actually, if you're an insider, this isn't quite true; but we'll get to 16(b) later.)

Stock Options

First, the history and terminology: Incentive Stock Options are the things that used to be called Qualified Options. Qualified Options were eliminated in the 70's; ISOs were authorized in 1981; and an incredible confiscatory hook was embedded in them in 1982. They are not to be confused with stock purchase plans, though they often are; when Osborne went belly-up, many people who thought they had options discovered that they were legally obliged to pay for worthless stock, with no option at all in the matter. We have an ISO Plan in operation now; if we set up a stock purchase plan, no one will be allowed to be confused between the two.

Under an ISO Plan the Board of Directors issues options which allow you to buy the stock at a set price (called the exercise price). If everyone follows all the rules exactly, you can exercise options at any time within a period specified in your option agreement; after waiting a while, you can sell the stock and pay capital gains tax on the difference between the price you paid and the price you sold for (see under Warrants).

Here are the most important of those rules that must be followed. First, the exercise price of the option must be at least the Fair Market Value of the stock as of the time when the option was issued; you hope, of course, to exercise at a time when the value is much higher, and to sell while it remains high. The option must expire within ten years after it's issued. Simple enough? Then remember that if the optionee owns 10% or more of the existing stock, the price must be at least 110% of fair market value, and the expiration must be no more than five years.

To exercise an option, send a check to the corporate Secretary (me) with a covering note to explain that you're paying this money to exercise options. We'll promptly record the sale of stock and issue a certificate. Please don't just mail or wire money to the company in the hope that we'll guess what it's for.

The time when you finally sell the stock must be (a) at least two years since the option was granted and (b) at least one year since you exercised the option. The recent change in the capital gains holding period doesn't affect (b). You also have to exercise the options First In First Out, and you can't get around this by getting the company to cancel old options before they expire. The reason for this is that a drop in the price of stock might tempt a company to cancel a lot of old, expensive options and issue new ones at the current, lower price; this is far too nice to the employees to be legal. (It's common practice to have a Vesting Period for options. This means that people can't exercise any of the options for (say) one year after the option date; then they can exercise up to (say) 25% of the option each year. The idea is to keep those bums from exercising their options and going off to take another job before the company has extracted full value from them. This is an industry norm, not a government requirement. Autodesk's vesting periods will conform to industry norms no more closely than the rest of our practices do.) That's all the important restrictions and traps, except for Alternative Minimum Tax and Section 16(b); we'll get to them later. So what if one of these rules is violated? Then [soundtrack: the Empire's theme music] the option becomes *disqualified*. When you exercise it, you become immediately liable for plain income tax (not capital gains) on the difference between the exercise price and the fair market value when you exercise. In the previous example, you exercise a $.10 option when the stock is trading for $5.00; whether or not you sell, you have income at that moment equal to $4.90, on which the federal tax could be up to $2.45. If you ever sell, your gain or loss will be computed against that $5.00 value rather than the $.10 exercise price. But what if the stock isn't registered, and it's illegal (as well as impossible) to sell it? No problem; the government will be glad to take your house if you can't raise the money to pay the tax on your non-existent gain.

If this happens to you, though, all is not lost: the company gets a tax deduction.

Alternative Minimum Tax

Important: Though most people don't have to pay any tax under this law, it appears to be necessary to file Form 6251 if you exercise any stock options, just to prove that you don't owe extra tax. So read this stuff. Alternative Minimum Tax is aimed at people who have a large dollar volume of capital gains, accelerated depreciation, option exercises, and intangible drilling costs (I'm not making this up).

Here's the theory behind AMT. Certain types of income get special tax preference, for reasons which are clearly

in the National Interest. This results in some people paying low taxes, which is clearly not in the National Interest. Therefore, the government gives subsidies and takes them back again, which is in the National Interest.

If you have these special types of income, here's roughly what you do. Compute your taxable income, and compute the tax on it. Remember this number. To your taxable income, add back some of the deductions you took: accelerated depreciation, intangible drilling costs, 60% of capital gains, and various itemized deductions. Also, if you exercised any Incentive Stock Options in the year, add the difference between the exercise price and the Fair Market Value when you exercised them (the $4.90 in our example, which you thought you didn't have to pay tax on because the options weren't disqualified). Subtract $30,000 (single person) or $40,000 (joint return). Take 20% of that. If this exceeds your normal tax, this is what you pay; otherwise (the great majority of cases) you merely report it and pay the normal tax. Stock options were included in this calculation under the Tax Equity and Fiscal responsibility Act of 1982. The equity of taxing people on a gain which another part of the law says is not to be realized for at least another year (and yet another says can't be realized without going to jail) is self-evident, so I won't explain it. What makes it especially equitable is that while a passive investor pays 20% on his gains, an entrepreneur is allowed to pay up to 40% (plus Uncle Deukmejian's cut). Comments in an earlier edition about rates going up to 52% were wrong, based on an uncritical acceptance of an expert's opinion.

This cloud, too, has a silver lining: if you're paying AMT, then the effective tax rate on any additional ordinary income is down to 20% until you get so much that you're out of AMT land again. This can be significant if you're liquidating assets to get enough money to pay AMT. If you need to understand this in more detail, stop by sometime when you have an extra hour or two, and I'll be glad to give you a quick outline of the situation.

Publicly Traded Stock

If the company succeeds, we would like to be able to cash in sooner or later. The normal way is to register the stock so that it can be sold on the open market. That, of course, is what we're planning now. Therefore, it's a good idea to know what to expect from a public offering. In order to register some of its stock with the Securities and Exchange Commission for public trading, a company spends $200,000 to $500,000 on lawyers, accountants, printers, etc., plus a few months of the management's time. If the deal falls through (which is quite possible from causes outside anyone's control), much of that money has to be paid anyway and is a dead loss.

The financial effect of the public offering is, to oversimplify, that the underwriters buy a few million dollars' worth of stock from the company and immediately sell it to the public at a profit. The buyers can then sell the stock to each other at ever-increasing prices, we hope. The part of the company that's sold in the offering varies widely, but might typically be 25%. The price is substantially higher than venture capitalists would have paid for the same percentage of ownership; a factor of two is often mentioned.

So now that the company is public, we can all start selling our stock to the public, right? Wrong. The underwriter may allow the existing stockholders to sell some of their stock at the same time that the company sells its stock, but the amount is limited by the public's desire to keep the founders a bit lean and hungry so that they'll make some further profits for the public before going off to lie on the beach. All the old stock that isn't sold then (the majority of the company) is still unregistered and still can't be traded easily.

Over the next few years the founders can dribble their unregistered stock onto the market under Rule 144, provided that the company keeps up with all the SEC's reporting requirements. The essence of this rule is that

anyone who has owned his unregistered stock free and clear for two years can sell through a broker in the public market, subject to a limit of 1% of the company (or a formula involving average trading volume) per three months. A person who is not an insider can forget about the 1% rule after owning the stock for three years.

As I understand it, people who own large amounts of stock, as well as officers and directors of the company, fall under Rule 144 restrictions even in trading registered stock.

Three years after the company goes public, it can file a simple little form ($20,000) to register all its stock. In the interim the stockholders could get some more stock registered by persuading the company and the underwriters to make another public offering (another quarter million dollars of expense, shared by the selling stockholders) in which a larger amount of their own stock is sold.

More practically, 90 days after going public, the company can make a simple filing (S–8) which will automatically register any stock that people get later by exercising stock options. Under this system a person who exercises an option for cash right after the effective date of the S–8 filing may be able to sell the stock immediately (though with terrible tax consequences), while someone who exercised earlier has to wait for two years under Rule 144. Nonetheless, this is a very good deal for the employees.

Insiders and Rule 16(b)

The first rule about insiders is that there are stiff penalties for trading on inside information. If you know that the company is about to buy a half-interest in IBM, and you think that will raise the price of the stock when it becomes known, you'd better not buy chunks of the stock to profit from the rise. This rule applies not only to officers and directors, but to anybody who has access to interesting information that isn't public yet. To help people resist temptation, the SEC requires all officers and directors of the company to report all their transactions in its stock. These reports are a matter of public record for every busybody and corporate gadfly in the country to study.

If two rules are good, three are better; therefore, the SEC isn't content just to levy fines for trading on inside information and to require reporting of transactions. It's also illegal for insiders to profit in any way, though in perfectly good faith, on short-term transactions in the company's securities. This is the egregious Rule 16(b): if an insider buys and sells a security in any six-month period, he must hand over his profit to the company! In any doubtful case this rule gets the most unfavorable interpretation possible; for instance, you take the losses and the company takes the gains, and you can't balance off losses and gains in a six-month period.

It's not entirely clear to me who is an insider under this rule, but it seems to apply to more than just the officers and directors. In fact, I have the impression it's not clear to anybody.

What if the company fails to claim the profit from an insider's trading? Then any stockholder can require it to do so. Remember the bit about the reports being public record?

Of course, none of us would trade on inside information or do short-term trading in the company's stock (much less sell it short, which is also illegal), but the rule can make real trouble. Exercising an option, for instance, counts as a purchase of stock; so don't sell any within six months before or after—there goes the idea of selling some stock to cover AMT, unless you exercise in the first half of the year. (It's said that there's even a way for an exercise to count as both a purchase and a sale, causing instant confiscation, but this appears to be just a trap for people without lawyers.) Or suppose you buy some stock, and three months later General Motors

comes along to buy the company—bye bye, profit. Remember, though, this just applies to insiders, if you can find out what an insider is.

This note has taken a fairly negative tone in places; in writing about regulatory matters one's attitude varies from heavy sarcasm to blind fury. It's as well to remember that people go through these things every day and come out with large bundles of money at the other end. If we can ace out Computervision, Autotrol, and IBM, not even the SEC can protect us against succeeding.

Why Lisp?

As the release of AutoCAD 2.1 loomed closer, we were somewhat diffident about unleashing Lisp as our application language. This was at the very peak of the hype-train about expert systems, artificial intelligence, and Lisp machines, and while we didn't mind the free publicity we'd gain from the choice of Lisp, we were afraid that what was, in fact, a very simple macro language embedded within AutoCAD would be perceived as requiring arcane and specialised knowledge and thus frighten off the very application developers for whom we implemented it.

In fact, when we first shipped AutoCAD 2.1, we didn't use the word "Lisp" at all—we called it the "variables and expressions feature". Only in release 2.18, in which we provided the full functional and iterative capabilities of Lisp, did we introduce the term "AutoLisp".

AutoCAD Applications Interface

Lisp Language Interface
Marketing Strategy Position Paper

by John Walker — February 5, 1985

Lisp?!?! Why the Hell did you pick the most arcane, obscure, and hopelessly-rooted-in-the-computer-science-department language in the world for an AutoCAD programming language?

Over the next six months, all of us will have the opportunity to answer this question. There are very good reasons why we chose Lisp as the initial language to attach to AutoCAD: I'll try to explain them herein. However, there is an important point we don't want to lose track of: the built-in Lisp interface we're providing is only the first in a series of Applications Interface products, allowing AutoCAD to be operated by application programs written in all major application languages. I anticipate interfaces to FORTRAN, compiled BASIC, C, and Pascal being available over the next 12 months.[184] Thus, Lisp is the language we sell for small applications—we are not offering it or suggesting it for major programming projects: that will be addressed by the other language interfaces, which permit a software vendor to attach their program to AutoCAD in its native language.

But back to Lisp. The following is my reply to "Why Lisp?".

Lisp is the preeminent language in the field of Artificial Intelligence, and has been for over two decades. Many of the most complicated programs ever written have been written in Lisp. Lisp is far from an esoteric toy of

[184] Alas, this was not to be. The memory constraints of MS-DOS have so far precluded this oft-requested feature.

computer scientists: a system called NAVEX, written entirely in Lisp, will soon be ensuring that the Space Shuttle reaches the runway. Expert systems implemented in Lisp will be a central part of the Space Station environmental and energy management systems.

Lisp is ideally suited to the unstructured interaction that characterises the design process. Unlike programming languages such as C and FORTRAN, which force one to organise a problem entirely before programming, Lisp encourages exploring various approaches to a problem interactively, exactly as CAD helps a designer.

No other major programming language can so easily manipulate the kinds of objects one works with in CAD. As opposed to numerical programming, CAD constantly works on collections of heterogeneous objects in variable sized groups. Lisp excels at this.

Because Autodesk's implementation of Lisp is completely interactive and provides on-line debugging facilities, Lisp is among the easiest of languages to master. Because the response to all changes is immediate, programs may be tested as easily as with an interactive BASIC interpreter.

Finally, the compelling reasons which make Lisp the language of choice for large applications are forcing the design of computers optimised for Lisp. Machines from Symbolics and Texas Instruments are already on the market. This technology will be crucial to high-performance systems of the late 1980's and 1990's. By moving CAD in this direction, Autodesk is positioning your applications to take advantage of this development.

So there!

Number One

As 1985 wore on, it became clear that we were on the verge of achieving our goal of having the largest installed base of any CAD company. I wrote this copy in March, 1985 to define a campaign around our large installed base. In modified form, this copy was used in a full-page advertisement and in our "Number One" company brochure. This was written with the intent of being used with the Apollo 17 full-Earth picture, and that's what we ended up doing.

This is the original draft of the copy, which is a little more hard-hitting than what finally ran.

Number One

by John Walker
March, 1985

In November 1982 we introduced AutoCAD, the computer-aided design and drafting program for personal computers, and said that AutoCAD would become the standard for CAD worldwide.

By March, 1985, we have shipped more than 17,000 copies of AutoCAD, making it the most widely installed and used computer aided design system in the world; micro, mini, or mainframe.

When we developed AutoCAD, we believed that the personal computer would rapidly become the core of the engineering workstation—a general purpose tool which assists the engineer, architect, designer, or drafter in all aspects of their work. We believed that AutoCAD could deliver mainframe CAD power as an essential part of this workstation. We believed that we could bring mainframe CAD to the personal computer without giving up the features and capacity which are the key reasons to use CAD in the first place. We believed that by making our system a fully open architecture and assisting others who wanted to build products around AutoCAD, hundreds of vertical market applications would be developed by those who shared our belief in the potentials of this market.

Very few of the traditional CAD vendors took us seriously. They looked upon the personal computer as something which might be able to do word processing or pie charts, but not serious design. They believed that if CAD was done at all on the PC, it would be done with limited-functionality programs for specific applications, "serious, general purpose CAD will always remain the province of the mini and mainframe".

We were right. They were wrong. Now they take us very seriously indeed.

And well they should. More than 17,000 users have already discovered that they can do serious, professional design work on the personal computers they already own. More than 1,000 dealers, systems houses, and OEMs

worldwide have discovered that computer aided design isn't an esoteric product for the Fortune 500, but an everyday tool as fundamental to people who draw as a word processor for people who write.

Our strategy to make AutoCAD the standard is working, and our commitment to this strategy is expanding. AutoCAD is continually being expanded and upgraded; our next upgrade will provide 3D visualisation, curve fitting, and a macro programming capability.[185] AutoCAD runs on over 31 personal computers, with more being released on a continuing basis. Vertical market applications such as AE/CADD, the professional design tool for architects, add to AutoCAD specific solutions for design professionals. CAD/camera, our expert-system based auto-vectorising system converts paper drawings to CAD automatically, and at $3,000, costs less than 5% the price of competitive systems. Our AutoCAD to mainframe translators allow integrating AutoCAD with large scale CAD systems including CADAM, Intergraph, and Computervision. And AutoCAD is available in French, German, Swedish, and Italian editions, with Spanish and Japanese scheduled for release soon.

If you design as part of your work, or draw, or your company designs or draws, you owe it to yourself to see what has made AutoCAD the CAD standard in so few months. If you own a personal computer, you already own the most expensive part of a professional CAD system. Just by adding the $2,500 AutoCAD software, you can immediately share the benefits that owners of million dollar CAD systems have been enjoying for over a decade.

AutoCAD. Number one—for a lot of very good reasons.

[185] This was release 2.1, which was shipped in May of 1985.

Prime Time

There was a feeling that as we became a public company and sought to expand the general awareness of our company and its products, we should retain a professional public relations firm. We hired "Primetime Publicity and Media Consulting Corporation", whose president, Reed Trencher, arranged for a writer to meet with numerous people in the company to produce a company profile.

This is what we got instead. Monuments stand alone. No commentary is necessary other than to explain the typography. Comments I wrote on the original document are in type **[like this]**. *Underlining and boxes around particularly trite and silly things were also my contemporaneous annotations. Footnotes were added at this writing. All of the misspellings were present in the original. Yes, they really did misspell the company name at every occurrence. Following this document is a memo I wrote expressing my reactions to this presentation of our company.*

AutoDesk

May 8, 1985

"We have large ears[186] turned to what customers and dealers need," says John Walker, President and Co-Founder of AutoDesk, Inc. "We make things people want". It is this philosophy—functionally unique among software companies—that has propelled AutoDesk to become one of THE success stories of the computer age.

An overstatement? Consider that AutoDesk revenues went from $14,500 in an abbreviated 1982 year to $10,000,000 last year. They have already cornered the personal computer market with AutoCAD, their brilliant Computer-Assisted Design program. Chosen by an international panel of computer journalists as the "Technical/Scientific Software of the Year" in 1984, AutoCAD provides a virtually-limitless palate **[Palette. Palate is part of your mouth.]** of drafting and graphic design tools for "anyone who draws". For the draftsperson, AutoCAD provides the freedom, scope, and power much like the way a word processor supports a pencil-pushing scribe.

Another AutoDesk breakthrough is price. The first hand-held calculator came out in 1961 at a cost of $29,000, while today more capabilities are available in a unit costing less than $20. **[What does this have to do with anything? Second, I'm not aware of any handheld calculator on the market in 1961 at any price.]** For under $10,000,you can buy AutoCad plus a state of the art IBM PC and equally powerful plotter,

[186] And some of us can wiggle them.

and you'll be getting a graphics design system that does 90% of what can be done on a main fram[187] system costing $500,000. But these are only statistics:[188] AutoDesk Incorporated are a group of people who have defied the "experts" and are now sitting on the crest of a wave that is sweeping in a new definition of business in America.

Their story is real, exciting, and challenging, like their roots: they are the Sixties Generation. After a decade of post-Vietnam invisibility, the new exemplars of the great entrepreneurial spirit are leading a shift in the way business will be done in the Nineties. That spirit is one of the most important reasons for the success of AutoDesk. No one says "We don't do things that way here". One difference is that competition within AutoDesk is constructive instead of destructive. "There is one ego in the company," said Director of Marketing Maryanne Zadfar, "the company itself. All of us are commited to each other and the company."

Communications Manager Sandra Boulton likens their operating style to Japanese management techniques, in which everyone's ideas are considered. This enfranchisement of the individual is the heart of AutoDesk; it is the way their products are designed. This assures that the programmers have the freedom to produce in their areas of highest expertise. And in their own individual style. **[This is a sentence?]** Many **[Come on]** of the employees work at home, communicating by modem and telephone, coming in for weekly conferences. Each person works on a particular part of the program, and then only at the end are all the parts be pulled together.

For those AutoDesk employees who work at the office, headquarters is a new, light, airy, office building in Sausalito, minutes from the Golden Gate Bridge, yards from San Francisco Bay, nurtured under the spiritual aegis of Mount Tamalpais **[Oh come on!]**. Those who do work at the office maintain their identities as personally as the established business community doesn't. **[What the Hell does this mean?]** Suits, regualtions, and titular respect have very little significance in the AutoDesk environment. It was this sort of unorthodoxy of style that stopped the venture capitalists from investing at the start-up. These "experts" asked about peoples' shoes, and questioned the length of their hair. They couldn't see that unorthodoxy—especially the people-consciousness—is the key to AutoDesk's flexibility. They could not understand that more important than structural formality was the enlistment of people who believe in themselves. **[Utter bullshit. VC's don't care about this, they had wrong ideas about the business!]**

The 106 who work at AutoDesk are individuals, **[What is the alternative?]** people who were likely considered rabble-rousers at least once in their lives. People whose attitudes are reflected in such traditional corporate heresies as "I don't like it" or "This doesn't make sense" or "There's a better way". The policy at AutoDesk is that there are virtually no regulations; the employees are "part of the rock". They all can buy stock in the privately-held company, so it's no wonder that "everyone is 100% behind every product that goes out the door".

AutoDesk was founded in late 1982 by fifteen engineers who pooled their own money—all of $59,000—and personal integrity to produce software that is literally changing the way many industries do business. Thought[189] computers have been used in drafting work for a number of years, it was basically only for those who had access to hundreds of thousands—indeed millions—of dollars of mainframe computer equipment. AutoCAD, the program responsible for the AutoDesk meteor, gave virtually the same tools to the owner of the every day personal computer for only $1,000. Individuals, small business, divisions of giant corporations all could make a quantum leap. **[This was tired in 1950!]**

In the personal computer industry for which so many had announced a premature death, AutoDesk has cre-

[187] As opposed to a secondary fram.
[188] Albeit, wrong ones.
[189] I think he meant "though".

ated such revolutionary software that it is actually forcing change **[Oh wow!]** in every industry that involves design, drafting, and drawing. Architects, engineers, contractors, electricians, mechanics, artists, doctors, soldiers, sailors, treasure hunters, teachers, firemen, landscapers, cinematographers, bobsledders, cooks, as well as designers of cars, stained glass windows, underwear, tennis shoes, sports cars, wheelchairs, contact lenses, bobsleds[190] and the face of the new Statue of Liberty **[Huh? I wasn't aware it was being redesigned?]** have all used AutoDesk Inc. products. Many firms are being forced to computerize their operations in order to attract the top design school students. **[I don't think that's how it works, folks.]** Other firms are requiring applicants to have 1–2 years experience with AutoCAD, **[Cite one example.]** and the program has been out less than three years.

It's because the AutoCAD system is synonomous with productivity **[Bullshit]**. What used to take days can be done in hours. AutoCAD frees the designer from the drudgery of repetitive, mechanical chores. In laymen's terms, AutoCad draws, and edits drawings, on a computer screen. These drawings can be manipulated in all sorts of clever and designer-practical ways. Not only does AutoCAD save countless hours of the most boring aspects of design work, it increases the accuracy of the calculations and provides a range of user-customized options previously unavailable on a personal computer.

As powerful as the system is, AutoCAD is simple to learn; a drafter with no computer experience can be proficient with AutoCAD in two weeks. And because AutoCAD is compatible with the major management systems like Lotus, Symphony, and Framework,[191] accurate and detailed design renderings can be integrated to support complex reports. AutoDesk also offers a variety of useful options to meet the particular needs of clients in their specific tasks and in the interface of their work with larger computer systems.

In addition to the extras offered by AutoDesk, the AutoCAD design system has already attracted more than 100 third-party software from outside sources to expand and customize the program for professions as diverse as theatrical lighting **[wrong]**, land surveying, and hydraulic network analysis. So AutoDesk customers are not only pleased, they are enthusiastic. And with good reason: AutoDesk implements user suggestions in redesigning their programs. When AutoCAD was upgraded from version 1.4 to 2.0, the number of callers with questions dropped 95%. **[Prove it.]** Former AutoCAD user **[Sounds like he abandoned it.]** and now AutoDesk engineer Lance Kemp restates the company's theme: "we give what our end users want."

AutoDesk is not a one product company. Originally, AutoDesk was the name of a program for a desk organizer written in 1982 along with a number of other software products. They **[Who.]** decided to put their resources behind AutoCAD. Will they try to resurrect the AutoDesk program? "We never look backwards." says Marketing Director Zadfar. Indeed, AutoDesk has two major releases in the Summer of '85: 3D Level 1 and CAD/Camera. The 3 D is actually 2 1/2 D, a program for providing three dimensional visualization of a design using either hidden lines on a "wire frame" schematic.

Cad/Camera allows a designer to enter existing diagrams **[drawings]** into the computer for editing and enhancement. This software breakthrough **[Bullshit]** means that the diagram on a piece of paper can be electronically translated into information that the computer can understand. Then with AutoCAD, the designer can manipulate the original image to whatever specifications s/he[192] wants, combine it with other images, and then reproduce with unparalleled accuracy and detail. Instead of spending twenty-four hours hand-tracing every element of a

[190] Went around a bend and came back a second time.

[191] Huh?

[192] Note to philologists. This is an artifact of the Marin County airhead version of an oddity of the late Twentieth Century called "non-sexist writing" which attempted to linguistically divorce women from the family of mankind. With the appearance of artificial intelligence, this trend led to the construction "s/he/it", which rapidly passed from the vogue taking the whole movement with it. The term is still heard in Texas but has a different meaning and is probably unrelated in derivation.

schematic drawing, with CAD/Camera the job can be done in two hours.

Integrity, intelligence, dedication—foundations in American history **[Huh?]**—are the resources of these people whom **[Whom?]** some labelled a bunch of left-over[193] hippies. While they are not accepted by the establishment whose very structure and purpose they challenge, it is of little consequence **[They sure as Hell better be before 2 July!**[194]**]**. AutoDesk has combined the business acumen with social values, and creating a dynamic deeper than their impressive statistical bottom line **[Huh? What? Is this a parody or what?]**. One of their users, a 23-year-old quadriplegic, sent a note of deep appreciation to AutoDesk, explaining that he had "signed" the purchase agreement as best he could, with an "X" drawn with AutoCAD. **[MAUDLIN! Do you have release & rights to this story?]**

But as successful as AutoDesk has been with their graphics-design systems, they make it clear they are not a CAD company.[195] Not shackled with hardware nor saddled with inventory, the people at AutoDesk have the ability to recognize a need, define the problem and create the solution. What comes next is being developed in the minds of bright and responsible individuals whose mutual badge is that they have accomplished what they were told couldn't be done. We might expect that what the Mustang was for Ford, AutoCAD will be for AutoDesk.

AutoDesk supplementary notes (5/2/85)

AutoDesk has thrown a scare into the design industry. Providing tools which translate into immense savings of time, money, and drudgery, **[What will I do with all the drudgery I save?]** AutoDesk is forcing drafters—architects, contracters, engineers, artists, and the like—to go electronic or they will be unable to compete.

The integrity of the AutoDesk operation is inherent in every operation: their advertising is completely—and proudly—truthful, and unlike many in the computer industry, AutoDesk reports its sales figures regularly and accurately.

Projected revenues for 1985 are $10,000,000. **[Wrong!** *Furthermore,*[196] *I don't think we're publishing any projections!***]**

AutoDesk has training centers around the country and in Europe. **[Hein?]** Their attitude is so contrary to the "sell-'n-see-'ya" indemic to the computer industry that they offer dealers a free program, simply for learning how to use the software.

[193] Yeah, right-on today, left-over tomorrow. That's life.
[194] The closing date of the public stock offering.
[195] Heaven forbid.
[196] Adds Dan Drake.

Reply Time

The Primetime Article

John Walker — May 14, 1985 22:41

I grasp for adjectives and fail.

Is this how we want to present the company we have all worked so hard to build? A freak show which uses every hype word in the lexicon of the flack, or as a group of hard working people whose dedication to competence and quality have made a name for ourselves?

Let me list some of the pure bullshit words used in this odious document:

> "functionally unique", "THE success stories", "brilliant", "virtually limitless", "breakthrough", "state-of-the-art", "sweeping in a new definition", "exemplars of the great entrepreneurial spirit", "Japanese Management techniques", "new, light, airy office building", "spiritual aegis", "people-consciousness", "literally changing the way many industries do business", "Autodesk meteor", "quantum leap", "revolutionary", "forcing change", "synonymous with productivity", "unparalleled accuracy", "foundations in American history", "combined business acumen with social values", "dynamic much deeper than their... bottom line".

I won't comment on the numerous egregious misstatements, gloss-overs, and misperceptions in this cowpie of a company profile. The overt illiteracy of the writing and slipshod editing is self-evident, but perhaps the centered, consciousness three, holistic well-being bubbleheads at whom this piffle is aimed are post-literate exemplars of the New Age.

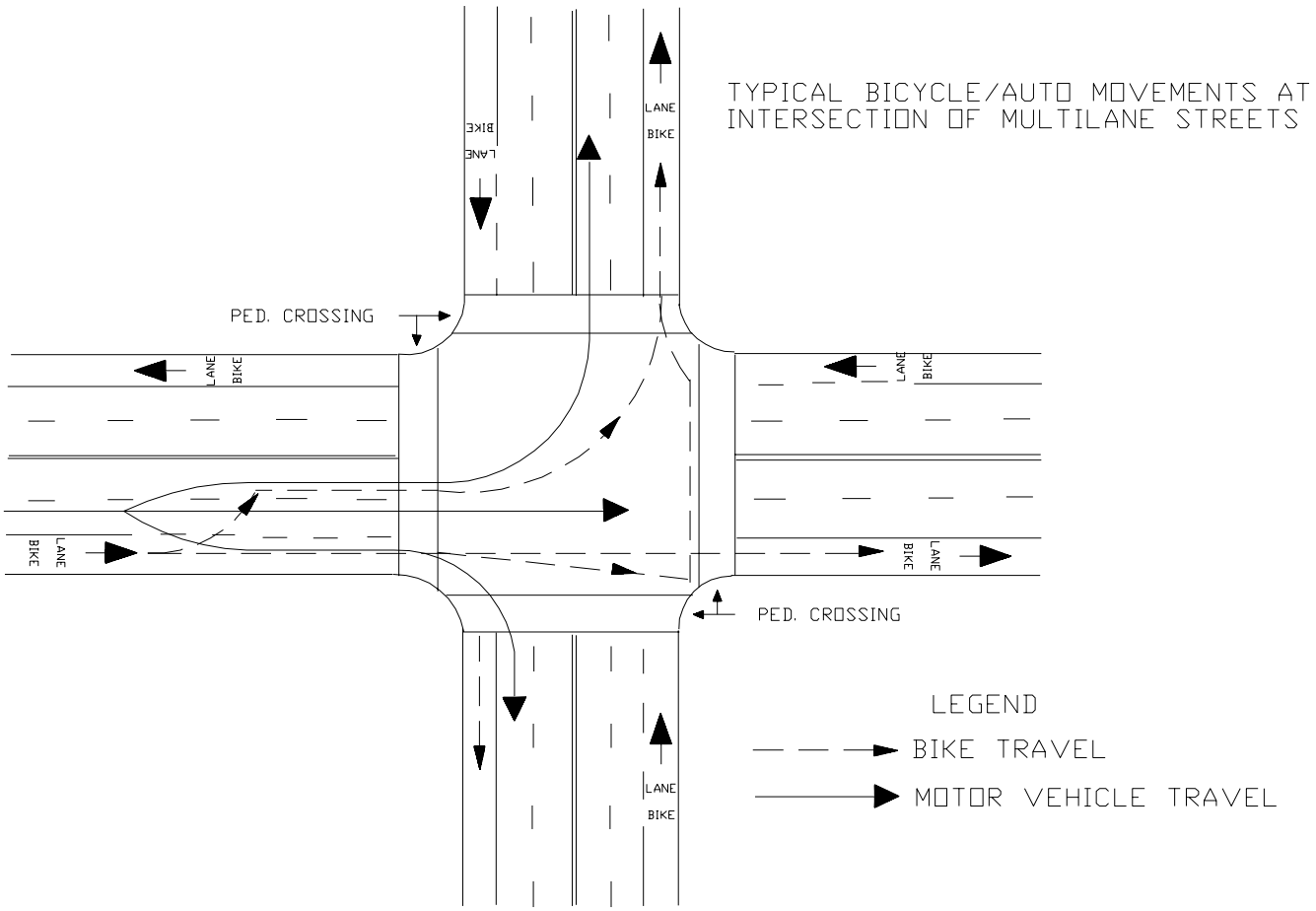
Would anybody who was impressed by this execrable effigy of our efforts be a likely customer of AutoCAD? Would we even want them to know we existed? If they stormed the building, we might have to lay in a supply of reality gas.

How can we respect the public relations judgement of anybody who could read trenchant prose like this and then submit it to a client? And if it wasn't reviewed before submission, what are their standards?

After meditating in the spiritual aegis of Mount Tamalpais, I must render my verdict upon this brie-dripping bastardization of our brainwork. I hope my attitudes don't forever align me with those "traditional corporate heresies" we disdain when I say,

"I don't like it".

Peter Barnett drew this representation of a street intersection based on a California Department of Transportation (Caltrans) specification manual. This drawing, done on a pre-release version of AutoCAD 2.0, uncovered several bugs in object snap, which was first introduced in that release. It has since been used as a sample drawing and plotter test.



Initial Public Offering

I did not enjoy writing the prospectus for our Initial Public Offering in 1985. Translating a clear statement of the company's goals and strategy into weasel words under a pressing deadline, in endless meetings filled with lawyers and accountants who argued with each other, billing the time to us pales, in my mind, with other avocations such as lying on the beach or juggling chainsaws.

Here is what we were trying to get across in the prospectus: the original draft that ended up devolving into the mealy-mouthed final document. It's the best statement I know of regarding where we were in 1985 and how we saw the company's future. The odd focus on "products" is because the big thing at the time was not to be seen as a One Product Company.

Business Section of Prospectus

Rough Draft 6 by John Walker — 4/16/85 01:27

General Background

Autodesk develops, markets, and supports a family of software packages which allow computer aided drafting, design, and drawing (CAD) to be performed on desktop microcomputers such as the IBM PC family.

CAD packages are used to produce drawings in such fields as architecture, civil, mechanical, and electrical engineering, surveying, facilities planning: any field in which information is communicated via drawings. A general purpose CAD package such as AutoCAD can make any drawing that can be made on paper.

The benefits of CAD are faster, more accurate generation of drawings, more efficient revision of drawings, the ability to use predefined symbols, eliminating time-consuming repetitive work and automatically assuring adherence to drafting standards. The benefits of preparing drawings on a CAD system exactly parallel using a word processor to write documents.

In addition, a CAD system such as AutoCAD maintains a database containing every element in the drawing. Users may attach information to objects in the drawing (for example, in a drawing of an office, a desk might carry its manufacturer, model number, date of purchase, price, and depreciation information). This information can be retrieved and modified from within the CAD program or sent to other application programs to prepare bills of materials, job costing reports, or inventory updates. A CAD system may thus be used as a "graphic database",

allowing design information to be taken directly from drawings, or conversely, allowing the presentation of design data in graphic form. AutoCAD was designed to make the integration of application programs for such purposes easy. Software suppliers serving structural engineers, surveyors, architects, and facilities planners, etc., can build applications based on AutoCAD, using it to accomplish the otherwise difficult tasks of graphic input, output, and editing inherent to their application.

Before AutoCAD, computer aided design was primarily done on mainframe and minicomputers, often with proprietary graphics hardware. Usually CAD systems were sold as integrated hardware and software ("turnkey") systems. With the introduction of the IBM PC and the many 16-bit desktop machines which followed, the basic desktop office computer reached a level of capability which allowed serious computer aided design to be done on the machine as supplied by the manufacturer. Thus, AutoCAD was introduced into an essentially vacant market: a *software package* for computer aided design sold separately from hardware and intended for use on existing desktop computers.

Additionally, AutoCAD was the first CAD package to support a wide variety of computer configurations. Today, AutoCAD runs on 31 different desktop computers and supports close to 100 graphic input, display, and output options.

AutoCAD's support of all major computers and graphics hardware is central to the Company's perception of the market and to its strategy. Exactly as portable, open-architecture operating systems such as Unix and MS-DOS have supplanted vendor-proprietary operating systems, and portable open-architecture networks such as Ethernet are supplanting those developed by computer vendors and sold only with their hardware, the Company feels that CAD customers will demand flexible CAD software which will run on a wide variety of hardware configurations and which can be expected to be available on newer, more powerful computer systems as they are announced.

AutoCAD is written in C, one of the most widely implemented and compatible computer languages for software development available today. Interfaces to operating systems, computer hardware, and graphics input, output, and display devices are completely separate from the main program, and may be changed without requiring alteration of the program itself. These design principles allow Autodesk to market AutoCAD on virtually any computer system which supports graphics and provides the C language. The C programming language is currently available on every serious candidate in the engineering workstation market, ranging from Apple's Macintosh to the Cray X/MP. This, combined with the proven portability of well-written programs written in C and the Company's experience in successfully moving its software from machine to machine, demonstrates that the Company can with minimal effort make its products available on any computer system it chooses as a potential market.

While CAD has been traditionally seen as a vertical market product (specific to one narrowly-defined industry), the Company feels that this has been more a result of the high price of turnkey CAD systems than the applicability of such systems. Just as word processors have become almost universally used by those who write and spreadsheet programs are widely used by those doing financial forecasting, CAD systems will soon be seen as essential by those who draw as part of their work as well as by full time drafters.

This large general market can be addressed only by those packages which require no special hardware, because such users cannot justify a special-purpose computer just for drawing. Instead, the drawing task will be done by a program running on their regular workstation, just as word processing and database software are used.

Autodesk's proprietary language translation utility vastly reduces the effort required to maintain foreign language editions of its products. Currently AutoCAD is available in English, French, German, Italian, and Swedish

editions. Spanish and Japanese editions are in preparation.

Product Strategy

The Company feels that the CAD component of an engineering workstation will succeed only if it meets the following criteria:

- It must run on the hardware the purchaser selects. The computer will not be primarily selected for the software; the software will be selected for the computer. Standard, hardware-independent software almost always supplants software tied to proprietary hardware.
- It must work with the other software components of the workstation. The workstation can deliver its promised productivity only if all the software forms an integrated design tool, as opposed to a set of distinct applications. The displacement of separate business application programs with integrated packages foreshadows this trend.
- It must be extensible and adaptable to the user's environment. No software vendor can anticipate the needs of all users, nor expend the effort to optimally customise the package for all applications. Instead, by providing users the appropriate tools, intelligent users or systems houses will do this in the field.
- The software must support third party vendors who wish to build applications based upon it. Open architecture systems usually displace vendor-controlled systems.
- The software must be general purpose and have no designed-in limits not imposed by the computer hardware itself. Users make a large investment in learning a package. They would rather spend 10% more time learning one package that meets all their needs than learn four packages which must be combined to solve the same problems.
- The software must communicate with mainframe computers and the corporate databases they contain. The engineering workstation does not exist in isolation. Designers work together, exchanging data, especially on large projects which can be managed only on mainframe systems. Also, large installed CAD systems benefit from the offloading of work which can be done effectively on desktop machines.
- The purchaser must be confident the software will continue to be available on new machines. A user makes a large investment learning a CAD package and adds to that investment with every project completed. Users must be guaranteed they will not have to learn a new system or throw away their drawings done on the old system when new hardware is selected.
- The system must be easy to learn and provide on-line assistance. Full time drafters have the time to attend multi-week training courses in CAD. Engineers and architects don't.
- The engineering and design business is of worldwide scope. To compete in the international market, a package must be available in the native languages of its users. People won't learn English to use a computer program.

The Company's products have been designed to meet all these criteria.

Products

AutoCAD

AutoCAD is a general purpose computer aided design and drafting software package. It provides the functions of a graphic editing system with attached database which form the core of every computer aided design system.

AutoCAD was designed to run on desktop computers, but does not contain any design limitations except those imposed by the present capacity of such machines. The designers' extensive experience in systems programming enabled the removal of limits in the software without degrading performance in the desktop environment.

For example, many early competitor programs imposed limits on the maximum size of a drawing which could be created or on the accuracy of the coordinates stored in a drawing. AutoCAD imposes no practical limit on either. Most early micro-based programs did not allow the user to modify the menus, or the help text, or design custom templates. AutoCAD allows all of these. The Company feels the success of AutoCAD to date expresses the market's verdict that these features are essential in serious design work.

AutoCAD is entirely written in the C programming language, is presently over 100,000 lines of source code (some small machine interface routines for some implementations are in assembly language).[197]

AutoCAD is microcomputer software only in the fact that it runs on microcomputers and that it exhibits the characteristics of ease of learning and use, good documentation, and user training tools one usually associates with microcomputer software. Its complexity, internal design, extensibility, and the general techniques used in its construction would normally identify the software as a mainframe or supermini package. As a result of AutoCAD's design, when presented with additional hardware resources such as higher resolution displays, faster processors, higher capacity internal memory (RAM), or larger discs, it automatically takes advantage of these resources and delivers their benefits to the user without software modification. AutoCAD's present internal design should easily accommodate the projected advances in these areas for the next decade. Thus, if run on a microcomputer, AutoCAD is a microcomputer CAD package. If moved to a minicomputer, it competes with other minicomputer CAD packages, and if moved to a mainframe, it becomes a mainframe CAD system. This, combined with AutoCAD's demonstrated portability, allows Autodesk to provide a compatible solution to the CAD industry on systems ranging from briefcase to room size.

The Company is committed to extending the capabilities of AutoCAD as well as the selection of hardware it supports. For example, the release of AutoCAD release 2.1 in May 1985 added three dimensional capabilities to the package, facilities essential for efficient use of drawings scanned by CAD/camera and for use with numerically controlled machines, and an initial version of what will soon become the full integration of the LISP language with AutoCAD. LISP is the first in a series of languages to be interfaced to AutoCAD, allowing users, OEMs, systems houses, and third party software developers access to the full capabilities of AutoCAD from their programs. Since LISP is the language of choice in artificial intelligence research, its provision within AutoCAD places AutoCAD on the leading edge of applying these techniques to the design process. The Company believes that the facilities these language interfaces will provide to application developers to be unique in the CAD industry, regardless of the scale of the system.

[197] AutoCAD doubled in size within a year after this was written, and by Release 12 in 1992, was more than a million lines of C.

Mainframe CAD Interfaces

While AutoCAD provides a total solution to the individual user or small office using CAD, users in larger corporations often wish to use their desktop workstations to develop drawings which are later combined with others' work on mini or mainframe CAD systems. Conversely, operators of expensive CAD systems wish to offload the large amount of routine work not requiring the power of the large system onto less expensive desktop machines. To meet these needs, and thus penetrate the corporate market for desktop CAD, Autodesk is developing a family of bidirectional translators which allow interchange of data with larger CAD systems. Translators for CADAM and Intergraph systems are presently available, with others under development. Autodesk believes that development of these translators is the key to establishing AutoCAD as the desktop CAD standard in major corporate accounts, and assigns a high priority to their development.

AE/CADD

An architectural design consists of drawings describing the structure to be built, plus extensive documentation provided to the contractor who constructs the building. AE/CADD[198] is an integrated design and drafting system designed especially for architects which automates drawing tasks and automatically builds the construction documentation directly from the drawing, guaranteeing consistency between the drawings and contractor information.

Driven directly from a digitiser template supplied with the package, AE/CADD automatically constructs walls from dimensions supplied by the designer, joins walls at intersections, breaks walls to insert doors and windows, and automatically draws stairs, plumbing fixtures, appliances, and structural details. Notes are automatically attached to markers in the drawing, and when the drawing is complete, AutoCAD's database link is used to automatically prepare the construction documentation describing the job.

AE/CADD allows an architect to make basic drawings much faster than with a general purpose CAD system, then eliminates the time consuming task of preparing the construction documentation. It is generated automatically from the drawing, preventing discrepancies which take time and cost money to correct in the field.

Autodesk plans to extend AE/CADD with additional templates to cover structural, mechanical, landscape, space planning, electrical, site planning, and plumbing drawings.

AE/CADD was constructed using the user-customisation features of AutoCAD. Written as a set of AutoCAD custom menus and symbols, AE/CADD may be installed on any machine which runs AutoCAD. The implementation of AE/CADD, accomplished initially by non-Autodesk personnel with access only to information provided to all AutoCAD purchasers, illustrates how AutoCAD can be adapted for specific application areas.

AE/CADD, sold with a suggested retail price of $1000, turns an AutoCAD system into a powerful design tool for architects.[199]

[198] Now AutoCAD AEC Architectural.

[199] This description was rewritten on May 12th, 1985. I've incorporated that version here.

CAD/camera

In order to take advantage of the many benefits of CAD, users with many existing manually-drawn paper drawings have had to manually transfer them into their CAD systems, in essence, redrawing them from scratch on the CAD system. The extreme cost of this labour intensive process has prevented most users from automating the filing and maintenance of their existing drawings when installing a CAD system. Rather, they have made new drawings on the CAD system, but maintained the old drawings manually. A system which automatically converted these paper drawings into CAD databases would be a great benefit to these users.

In addition, upon installing a CAD system, the purchaser must usually spend a great deal of time entering commonly used symbols and drawing details before being able to realise the full benefits of CAD. The ability to enter these symbols automatically for immediate use by the CAD system would save users much time and deliver immediate productivity gains.

Autodesk developed CAD/camera to satisfy both of these needs. CAD/camera allows users to automatically transfer their paper drawings to CAD databases. Taking an image scanned with an electronic scanning camera, the CAD/camera software package translates the scanned page to the vector form usable with CAD systems. Existing systems which perform this function are based on mini and mainframe computers and cost more than $100,000. CAD/camera, by contrast, runs on personal computers and is sold as a software package alone for $3000. When CAD/camera is run on an IBM PC/AT, conversion times for drawings range from 15 seconds for small symbols to more than five hours for complex engineering drawings. This is usually at least ten times faster than manually redrawing the drawings on a CAD system.

CAD/camera is implemented using rule-based expert system technology, which is responsible for its much greater price-performance, and its ability to run on smaller, less expensive computers. In addition, this technology allows Autodesk to continue to enhance CAD/camera, adding recognition of more complex drawing elements.

Databases created by CAD/camera may be directly read by AutoCAD, but CAD/camera may be used to generate databases for any CAD system. Its output format is fully disclosed by Autodesk, facilitating its interfacing with other systems. In addition, CAD/camera is entirely written in the C programming language, allowing it to be moved to other computer systems, including other CAD systems should Autodesk decide to do so.[200]

Developers' Tool Kit

As more and more graphics hardware comes onto the market, Autodesk plans to support it in AutoCAD to maximise the user's choice. The large installed base and rapid sales pace of AutoCAD makes it an important potential market for developers of graphics hardware. Autodesk's Developer's Tool Kit makes the union of these common interests less costly and time consuming to both parties. After evaluating a piece of hardware and concluding that support of it by AutoCAD would be beneficial to the Company and its customers, a Developer's Tool Kit may be sold to the hardware vendor. Using a manual specially written for use with the Kit, the hardware vendor can program a driver which allows AutoCAD to run his device. Since developers are usually more experienced in programming their hardware than Autodesk, this expedites the development process. After the driver is complete, is it certified by Autodesk's Quality department before shipment with AutoCAD. Autodesk retains title to the driver developed by the hardware vendor and has so constructed the Kit that it discloses no proprietary information. Autodesk charges a fee for the Kit which covers support costs

[200] This description was rewritten on May 12th, 1985. I've incorporated that version here.

in aiding the developer in using the Kit.

AutoCAD Applications Program

Autodesk actively encourages the development of third party software which works with AutoCAD and aids its use in vertical markets. Autodesk has established the AutoCAD Applications Program as a channel by which developers of such programs may communicate with AutoCAD vendors and users. The first AutoCAD Applications Catalogue contains more than 100 such programs. Autodesk derives no revenue from these third-party programs except that generated by additional sales of AutoCAD they engender. However, the Company believes that this Program is an excellent way to identify and qualify programs for possible acquisition, joint marketing, distribution, or licensing by the Company.

The Toilet Announcement

Getting screen pictures to print in the prospectus wasn't easy, either. In fact, nothing about the public offering was easy. However, difficulty shouldn't make one hesitant to break new ground and defy precedents. Dan Drake penned this press release in the midst of the prospectus drafting sessions.

For Immediate Release

Sausalito, California. May 10, 1985.

In what industry observers described as a radical and daring break with tradition, Autodesk Inc. announced today that the publicity pictures in its prospectus would not feature a picture of a toilet. The decision was announced following an extraordinary meeting of the Board of Directors.

"It's hard to part with an old friend", gibbered John Walker, president of Autodesk, emerging from the meeting which was held in the company's washroom, "when our whole success has been based on pictures featuring toilets, from the mini-apartment drawing in *PC World* to the giant North Sea oil rig poster. However, hard times demand hard choices. We at Autodesk are flushed with pride in our forthcoming public offering, and as we stand with one foot in the simpler world of private companies and the other in our mouth, we hail the dawn of the new day confident that our publicity will continue to bowl over the industry."[201]

[201] Alas, it was not to be. One of the screen shots inside the prospectus cover *did* include a toilet.

Marketing Strategy

Sales and Marketing

Introduction by John Walker — May 14, 1985 01:14

In keeping with Autodesk's overall strategy of delivering the benefits of CAD to a mass market, Autodesk's marketing strategy is to apply the time proven techniques of mass marketing to a product traditionally sold directly at high prices.

This strategy, unique in the CAD market, complements the technical benefits of AutoCAD. Its application allowed Autodesk to obtain its large market share in a short time. In addition to applying mass marketing techniques itself, Autodesk mobilises the sales forces of computer manufacturers, graphics peripheral manufacturers, and computer dealers through cooperative advertising, promotion, and appearance in numerous trade shows. Autodesk has a variety of innovative programs involving training, advertising credits, joint appearances at trade shows, and other incentives which encourage dealers and manufacturers to jointly market Autodesk products.

Autodesk supports its advertising with an aggressive public relations effort, combined with an ongoing program of seeking and arranging for the publication of articles in the trade press describing applications of AutoCAD in various industries. Autodesk makes a major ongoing effort to communicate with industry analysts and key decision makers, seeking to demonstrate the benefits of AutoCAD versus larger systems. Autodesk supports the development of tutorial materials and books based on AutoCAD. Finally, Autodesk has a major commitment to the educational market, offering support and incentives to institutions wishing to teach CAD, and encouraging the adoption of AutoCAD in their curricula.

To reach a mass market at a low cost, the Company has concentrated on two major channels of distribution: computer dealers and computer manufacturers. The Company's approach in promoting both of these channels has been to communicate the real advantages of selling AutoCAD to participants in both market segments.

Computer dealers who sell AutoCAD typically make more from the dealer markup on AutoCAD than the retail price of most of the software packages they sell. In addition, the AutoCAD customer usually buys a larger computer with more options (larger memory, floating point coprocessor, larger disc storage) and with graphics peripherals such as a digitiser and plotter. These options and peripherals are typically discounted less in the marketplace than basic microcomputers, so the dealer's margin on the overall sale is increased by selling AutoCAD systems. These larger margins and access to less competitive vertical markets usually more than repay the dealer's investment in learning to sell AutoCAD. The Company's policy of not selling directly to large accounts and not placing its products in discount prone national distribution channels serves to strengthen its dealer network and that network's loyalty to the Company and its products.

Computer manufacturers who sell AutoCAD gain access to vertical markets previously denied them and gain a tool which uses their hardware to best advantage. Because AutoCAD automatically makes use of the resources provided by a computer system, whatever competitive advantages a system may have (better graphics resolution, higher performance, larger memory, larger disc storage) are effectively utilised by AutoCAD. Thus in a crowded, highly competitive market, AutoCAD provides a computer manufacturer a product which dramatically illustrates the advantages of his product versus the competition, demonstrably promoting hardware sales. In addition, the manufacturer receives significant revenue from the sales of AutoCAD software, while encouraging the sale of larger, more profitable machines. AutoCAD provides access to vertical markets within which the

specialisation of a manufacturer may yield much greater results than in the general PC market. Computer manufacturers typically distribute AutoCAD through the same channels through which they sell their hardware; some manufacturers sell through their own dealer networks while others sell directly, mostly to large organisations.

The Company's longer term marketing strategy builds on the concept of AutoCAD as a general purpose tool which forms the central component of an engineering workstation. While AutoCAD by itself delivers compelling gains in productivity easily communicated and justifying its purchase, an AutoCAD user is a prequalified customer for a wide variety of additional productivity tools. These tools include predefined symbol libraries; a wide variety of engineering and design automation programs for such purposes as preparation of bills of material, job cost estimation, structural analysis, numerical controlled machine tool programming, and electronic circuit analysis; and materials intended for use with AutoCAD, such as templates, tutorial guides, and other self-teaching materials. Autodesk regards its large and rapidly growing base of customers as one of its major assets, and intends to develop and market additional productivity tools into this base. CAD/camera and AE/CADD are examples of additional Autodesk products which will appeal to significant numbers of AutoCAD customers, as well as encouraging new sales of AutoCAD. The company's large installed base also leads third party vendors of applications software which complements AutoCAD to approach Autodesk with joint marketing proposals. These products, qualified through the AutoCAD Applications Program, provide a continuing source of new products for joint marketing or acquisition by the Company.

In short, the Company's marketing strategy is to create a mass market for CAD, where no mass market existed before, develop channels of distribution to address that mass market, and build on its emerging position as the volume leader with additional products and services.

The Entire Prospectus

Kelvin Throop was infuriated by the prospectus drafting process. He suggested we can the entire mess and use this prospectus instead. We didn't.

The Entire Prospectus

Draft 1 by Kelvin R. Throop — 2/30/85 24:12

In the beginning CAD systems were overpriced, hulking boxes of hardware with the original nameplate pryed off and the name of some slimy greedhead stuck on.

Then came AutoCAD, a program that did all the same things on a PC for 5% of the cost.

Things got better. As they got better, we got richer.

Now's your chance.

Call toll-free 24 hours per day, (800) AI-STOCK. Visa/MC/Amex accepted.

Sleazy Motel Roach Hammer Awards

One of the most repellent parts of the public offering process was the extravagance of the "road show". Apparently investment bankers believe they can do their job better when consuming their firm's capital at an enormous rate on such things as first class airfares, limousines, $200 a night hotel rooms, and the like.

Now that Autodesk had obtained a large wad of cash, I was concerned that we would also start to go down the same road. This was my proposal to create an incentive system to keep that from happening. This was never implemented.

The Autodesk Sleazy Motel Roach Hammer Awards

By John Walker — June 22, 1985

It sure is expensive to travel, isn't it?

Having just survived the "road show" phase of the public offering process, I've just been reminded of the needless extravagance the travel establishment lavishes on expense-account corporate America. If the people who were doing this traveling were paying out of their own pockets rather than "the company's", I'll bet that hundred dollar a night hotel rooms and fifteen dollar dinners wouldn't be long for the world. Five minutes, say.

Now every growing company, especially those who have recently gone public and now have the world looking over their shoulders and watching their margins (sales less expenses), has to issue the Obligatory Let's Control Costs Memo and some utterly confusing policy which is destined to be ignored and end up in the circular file of history.

Autodesk was built on incentives, not coercion. The way to control costs is to make it pay. Henceforth, there will be a direct financial incentive to keep costs down. Those who travel on business have to fill out travel expense reports listing the direct costs of their travel. This form will be amended to add a calculation of the "sleaze factor" of the trip. Sleaze factor is defined as the number of days the traveler was out of town (one for day trips), divided by the money spent on the trip, exclusive of air transportation.

The accounting department will keep track of the cumulative sleaze factor for all people who travel. At the end of each month, the traveler with the highest sleaze factor (who therefore cost the company the least per day on the road) will receive a bonus in the next paycheck of $200.

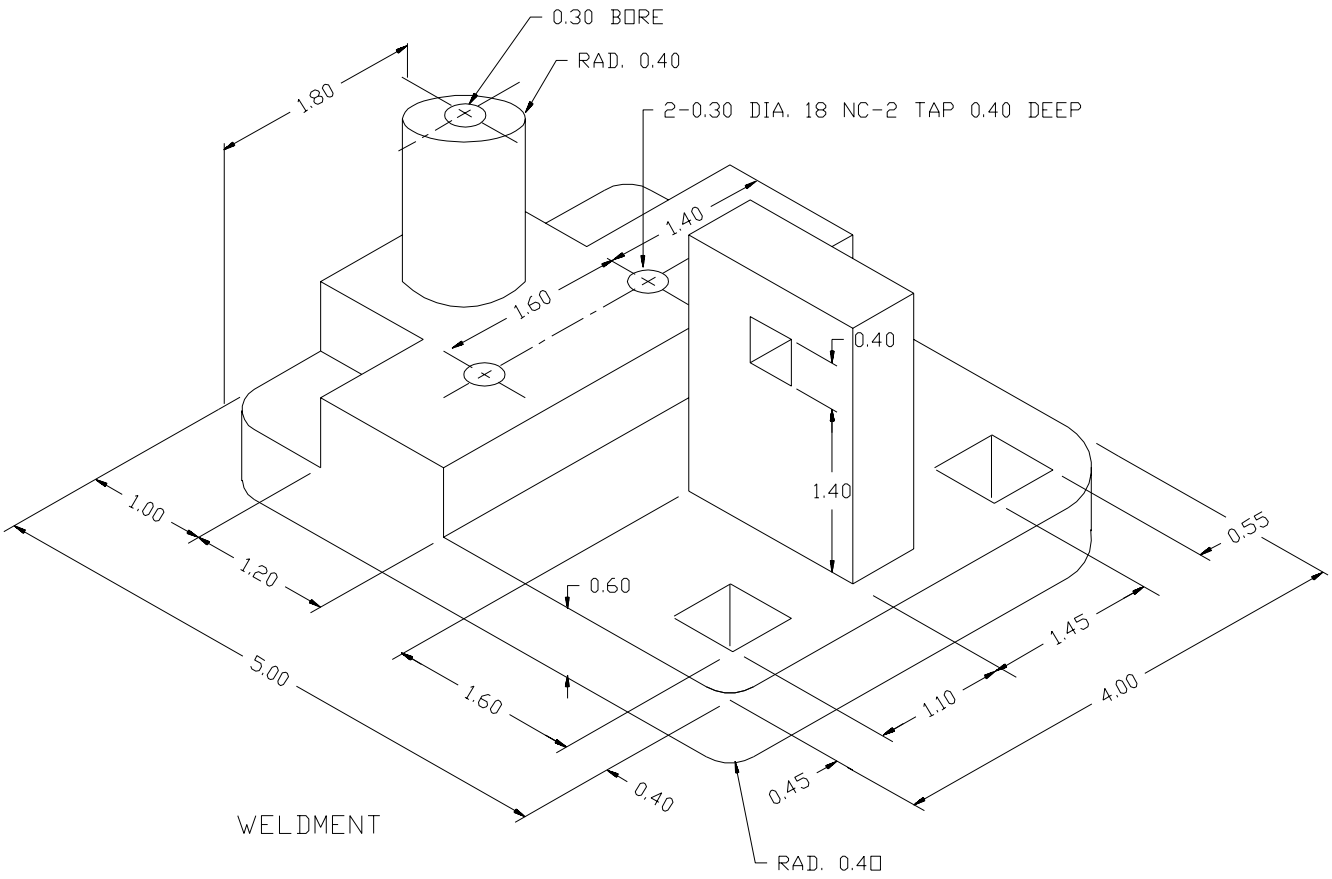
At the end of the fiscal year (January 31), the employee with the highest yearly sleaze factor will receive a bonus of $2500 in the next paycheck.

In addition, the person who turns in the lowest cost per day will be honoured at the next monthly meeting and presented the Autodesk Sleazy Motel Roach Hammer Award.

This award program is not totally fair. But then life isn't totally fair. Somebody who goes to New York repeatedly will tend to run up bills higher than one who frequents Akron. But then some say that New York is its own reward. But in any case, the point of all of this is to reward those who treat the company's money as if it was their own. It is, you know. Everybody here owns the company, either directly or as the holder of a stock option. If we keep the costs down and consistently turn in results that meet or beat the expectations of the

outside world, we can see the value of our company increase by a factor of 10 to 20 over the next five years. That is the goal, and if we achieve it, we will all be able to share the rewards of our work and the prestige of the company that we built together.

This notorious drawing was pulled from our sample drawings disc because it faked, by laborious manual methods, various features that would have been nice in the package but weren't there. Some of them still aren't in Release 9. The drawing was originally made by Peter Barnett in 1984, and was intended to illustrate the isometric grid and snap features in AutoCAD 2.0. The isometric dimensions were all hand-drawn, and the ellipses were made by differentially scaling a block containing a circle.



Surplus Value

With the public offering complete, many people who had been essentially broke the month before found themselves bombarded by those willing to help them solve the problems created by their (largely paper) "wealth". I thought it would be a good idea to pen an introduction to the investment world for people who had ignored it before in the hope that at least the most egregious fleecers and slimebags would be seen through. Other than details about taxes, which are dated, I wouldn't change a word today.

Surplus Value

Revision 3
By John Walker — July 1, 1985

Fins to the left,
fins to the right,
and you're the only bait in town.
Jimmy Buffett, *Fins*

When you sold stock in the public offering and your name appeared in the prospectus, you committed an act not unlike pouring blood in the water before taking a swim in shark-infested waters.

Whatever your financial situation may be, to those who read the 40,000 copies of the prospectus we paid to print, you "have money", and can be expected to be pursued by those who want to "help you manage it".

Look out.

I do not presume to suggest to anybody what they should do with the money they got from selling the stock. It's yours; you earned it. The only purpose of this note is to share some of my thinking about the question we now face: "what to do with the money". The thoughts herein are biased by my own financial situation and may be completely inapplicable to yours. I'd also like to share some words of warning about some of the predatory types who will soon begin to circle.

And of course, please assume that everything I tell you is totally wrong and "do not take any action without consulting with your own financial advisors".

Taxes

Well, we are going to be paying a lot of taxes this year. I think that this year I'm not going to be doing my own taxes. The proceeds from the sale of the stock will be considered a long term capital gain for federal taxes, assuming you sold stock purchased at inception. But remember that California has a three tier capital gain structure and that you don't get the lowest rate until you hold the stock for five years, so none of us will be in the lowest California bracket. If you exercised any options this year (and of course everybody did), you also have to calculate Alternative Minimum Tax (AMT), even if you don't end up owing any. And remember that California also has its own Alternative Minimum Tax, which will crank the effective California capital gains rate up to about 9.5% (don't complain: we all have to Do Our Part to contribute to the entrepreneurial renaissance in New Hampshire and Texas). So in any case, the calculation is going to be complicated.

Here are some random thoughts regarding the tax situation:

First of all, we can't wait until next April 15 to worry about the taxes. We'll have to make the next estimated tax payment on September 17. So a goodly part of the money you kiss hello on July 8, you will be kissing goodbye in September. You'll absolutely have to be able to make a quick shot at your 1985 tax liability and *make that payment*, because if you miss it, you can kiss something else goodbye. Estimated taxes are tricky, and there are several gimmicks which can help you keep the money in your hands for longer. For example, if your withholding plus estimated taxes for each quarter exceeds last year's tax liability, you don't have to make additional payments; you can just pony up the balance next April 15 and file the "hey, it's cool" form. But to do this, you'd have had to have made the qualifying payments last April and June. Did you? I sure didn't. Also, at the end of the year, the buggers will probably hit you for a deficiency because you didn't make estimated tax payments in April and June. You'll have to prove that your large slug of income didn't come until third quarter. Be sure you can.

Also, when calculating your taxes for estimated tax purposes, remember that you'll probably benefit substantially from income averaging this year. Don't overpay estimated taxes because you forgot this when making the estimate.

This probably doesn't apply to anybody, but I'll mention it just in case. If you have any long term capital losses (that Atari stock you bought when video games were going to the moon, the $800 gold coins, etc.) that you haven't realised, take the losses this year. You can offset long term capital losses dollar for dollar against gains, but you can only deduct $3000 of loss per year in excess of gain, so this year you can flush out all those unrealised losses. If you still want to hold the assets, buy 'em back more than 30 days later (the delay is to avoid a "wash sale", discounted for tax purposes).

Also, I don't think that anybody will have a significant excess AMT liability, but maybe your kid went to a painless dentist and you have some Intangible Drilling Expenses and are in AMT land. As long as your AMT exceeds your ordinary income, additional ordinary income is taxed at only 20%. So if you can discretionarily generate ordinary income (such as selling short term stocks at a gain, etc.), do it as long as your ordinary tax doesn't reach the AMT number. Conversely, if you're in a position of excess AMT, you want to put off taking any short term capital losses or deductible expenses (charitable contributions, etc.) because as long as you're in a 20% marginal bracket, Uncle is paying only 20%. If you can delay them to next year, you may be in a higher bracket.

Investments

All right, you've paid off all the bills, beaten the wolf back from the door to at least the porch steps, and you have some money left over. Now you're ready to talk to those guys who are calling you five or ten times a day to tell you what to do with it, right? Wrong. First, make sure you do the obvious little things, such as (if you haven't already done so) prepaying your IRA for 1986. You can earn the interest on $2000 ($4000 if married) tax free for a whole year by prepaying now. Next you have to think about "your portfolio".

Most of the paper peddlers who call you will consider your cash and tell you how to deploy it to "meet your financial goals". The cash will become a mix of investments which will be called "your portfolio" (as if you carried it around with you all the time—though in a sense you do by worrying about it). Unfortunately, they often ignore the other 90% of your net worth. What's that? Your stock in Autodesk. So remember that your portfolio is already invested 90% or more in a high-risk, high-tech company, so anybody who advises you to put any of your cash into similar stocks for "aggressive growth" is telling you to increase your concentration in this sector. What you probably want is to balance things by staying pretty conservative with where you put the cash, so be sure whoever is advising you understands the whole picture. The best book I've seen about portfolio balancing and evaluating different risk factors is called *Inflation-Proofing Your Investments* by Harry Browne.[202] I *do not agree* with much of the advice and specific recommendations given in this book, but the sections on valuing differing kinds of holdings (equity in a house versus bank deposits versus shares you can't sell) are very well written and easy to follow.

You will probably be contacted by people who call themselves "financial planners". There are two kinds of people going by this name. Some prepare a plan from information you supply, for a fee. Most derive their income from commissions on specific products they recommend and then sell you. 'Nuff said.

If you're looking to stash the cash immediately in a safe place that generates income, I'd recommend Capital Preservation Fund, which I have used since 1978 and with which I have had absolutely no difficulties. They invest only in US Treasury Bills, which are generally considered the safest investment in the world. They also have a fund which is free of both California and Federal income taxes. You can write checks on either fund. (Again, I'm not telling you to put your money there, and they may run off to Paraguay with it tomorrow. But if they do, I'll lose a lot.)

Before getting involved in any investment other than ultra-safe short term things like T-Bills, it's worth spending some time learning just what the rules are and what all this stuff they're trying to sell you is. My favourite introduction to the game is a book called *How to Buy Stocks* by Louis Engel.[203] I also like *The Only Investment Guide You'll Ever Need* by Andrew Tobias,[204] but I like this book a lot less than some people do and consider it mistitled. I'd read it for background, but not advice.

I have a lot of other references and information about investments. You're welcome to borrow any of them. Most are in my office.

Greed and Fear

> "When I hear the word 'culture', I reach for my gun."

[202] *Inflation-Proofing Your Investments* by Harry Browne and Terry Coxon, William Morrow & Co., 1981.
[203] *How to Buy Stocks* by Louis Engel in collaboration with Peter Wyckoff, Bantam Books, 1977.
[204] *The Only Investment Guide You'll Ever Need* by Andrew Tobias, Bantam Books, 1979.

Hermann Goering

"When I hear the word 'leverage', I reach for my coat and head for the door."
Dan Drake

"In investing money the amount of interest you want should depend on whether you want to eat well or sleep well."
J. Kenfield Morley

Dan Drake suggested that I add a section talking about the kinds of things to watch out for. Frankly I'm of two minds about this. Walker's first law of investing says, "If you don't *totally* understand it, ignore it". I cannot possibly give you enough information herein to make an intelligent decision, so I'll just concentrate on the lingo. But if you're unwilling to take the time to learn the game, I think you're better off not playing at all. Professional money managers have years of training, access to extensive libraries of research material, massive computer support systems, and full time analysts watching every piece of data. Yet few of them do better than random chance. If you intend to better them, realise you're going into a business venture and prepare to spend the time and effort a business requires.

What follows is Walker's acerbic, opinionated, *tour d'horizon* of investments.

With your money you can spend it on *stuff* or *paper*. *Stuff* includes BMW's, yachts, Big Macs, houses, and gold. *Paper* includes stocks, bonds, CD's, options, futures, options on futures, futures on options on futures on gold, etc.

Bonds are debt. You give somebody your money and they agree to pay you back someday (if soon, like 90 days, it's "short term", if not, like 30 years, it's "long term". Exercise: what does "intermediate term" mean? See, it's not so hard!), and to pay you interest at some percentage rate. Usually, the longer the term, the higher the interest. But the longer the term, the higher the risk, because if interest rates go up, the value of your bond goes down. Also, there's the risk that the issuer won't pay off, or may even stop paying interest. Issuers with tons of cash and a record of prudent financial management such as the U.S. Government get to pay less interest than fly by night operations like IBM. In general, the greater the risk, the higher the interest.

Stocks are equity. You own part of something. This can range from the telephone company to Autodesk. Generally stodgy old companies pay you a dividend in cash and are much safer. Utility stocks are the stodgiest of all and are very similar to bonds. Stocks in established, well capitalised companies such as Computervision or Union Carbide are much safer than wild-ass startups like Apple, Intel, Tandem, and Autodesk. This is because it is less painful to lose your money in good company.

All the rest are pinstripe Las Vegas in New York (or Chicago). The purpose is to have the most fun as you lose your money.

So what's leverage? Leverage is how you can lose or possibly make money even faster. Options (buying, not writing), buying stocks on margin, and futures are ways to obtain leverage. You can, by proper application of leverage, lose even more than you invested. Isn't that *neat*? (All right, this is somewhat unfair. Leverage, properly applied, can let you hedge illiquid assets and shift risks to speculators willing to assume them. Leveraged markets are essential to the efficient deployment of capital in a free market. See you at the track.)

And I could go on and on. This is really fascinating, and as one who has long been a market follower and player, I could go on for hours. But as I swore off all market playing when I started Autodesk, I'd rather not. It seems to me that it's a lot easier to make money than to multiply it, and for the moment, that's my focus.

Beyond the Lock-up Period

Since all selling shareholders signed a six month lock up agreement, sales of stock under Rule 144 are not an immediate concern. This is good, because all the people who will call you about their "restricted securities program" can be got rid of for at least six months. But 1986 will bring them out of the woodwork. Subject only to the constraints of the law, you will then be able to sell your stock through any broker willing to do the paperwork and abide by the rules.

You should be concerned, however, with sales of stock affecting the price. Remember that there are only 1.4 million shares out there. We hope that a large percentage of those will be in "strong hands", that is, long term holders. Thus the "float" or volume that actively trades may be quite small. As a result, throwing a block of 10,000 or 20,000 shares on the market may knock the price down significantly. As a result there may be an advantage in selling the stock through a broker who is a primary market maker in the stock, since they will sometimes have a better feel for how well the market can accept the stock and when is a good time to sell it. Initially, the market makers will be our underwriters. We hope to pick up additional market makers in the future (the more the better as far as the company is concerned), and we'll let you know.

Remember that brokers get a commission when they sell stock, and that people will be actively prospecting for this business. You will receive calls that begin "I have a buyer for 20,000 shares of Autodesk stock. If you're interested in selling, we can do the transaction, and since I have the buyer already, the price won't blip down". And when you agree to sell, he'll start looking for that buyer.

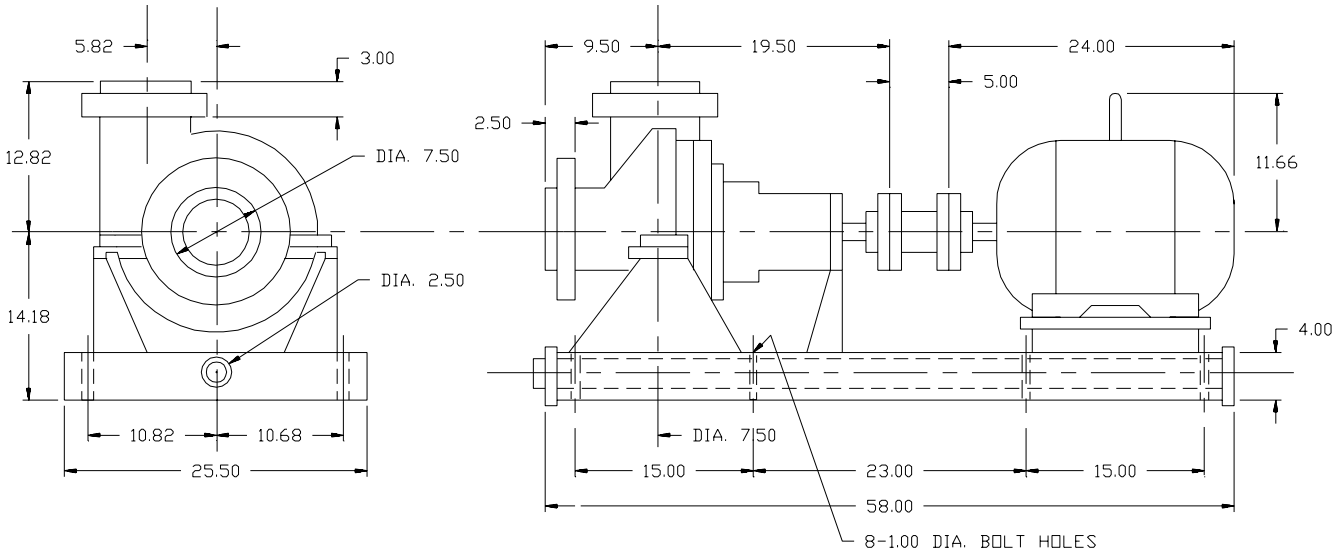
Anyway, watch out. From now on, assume you are a target. Not everybody who calls you up blind with a financial "opportunity" is a total sleazebag trying to loot the efforts of your hard work. I've made a list of those who aren't, and have already written the title at the top of the page. Now all I need is the first name. Let me know if you encounter one.

And remember, these guys can consume hours and hours of your time. Don't hesitate to be rude. I've found only one thing so far that gets rid of these guys immediately without overt hostility, and that's saying the magic words, "That's very interesting, but I'm totally broke". And now even that won't work.[205] Anyway, your time is your own, not theirs.

There's a reason they're called *broker*s.

[205] I was wrong. Amazingly, it *still* works pretty well. They're pushy, but not particularly smart.

Peter Barnett drew this pump in 1984 to illustrate mechanical applications of AutoCAD. It has appeared on the AutoCAD sample drawings disc from Version 2.0 to date.



Information Letter 12

This Information Letter was written for the first Founders' Meeting. The Founders' Meeting was held at Dan Drake's house in Oakland not long after the public offering had been accomplished. It was an opportunity to review the path we had traveled and the challenges we might face next. I wrote this the night before the meeting, racing down to the Marinship office at midnight in search of references in my bookcase there.

Autodesk, Inc.
Information Letter # 12

by John Walker
Revision 2 — Moon Day, 1985

> "I resolved to make such an attempt at 'clarification', fully realising that it would increase the size of the pamphlet and delay its publication; I saw no other way of meeting my pledge I had made in the article "Where To Begin". Thus, to the apologies for the delay, I must add others for the serious literary shortcomings of the pamphlet. I had to work in great haste, with frequent interruptions by a variety of other tasks."
>
> V. I. Lenin, *What Is To Be Done*, 1902

We did it.

In January, 1982 we got together and decided to build a software company which would become an industry leader. We agreed that our goal was to build a large, conventional, tightly-coupled company which provided all the services needed to become an industry leader. We all committed a major component of our time, and put at risk a substantial portion of our financial assets.

Today, Autodesk is one of the leading software companies in the world. Our goal was to build a company which would be one of the top five. By Mike Ford's analysis of the SoftLetter 100, discounting game companies and people who have collapsed since the list was published, we are about number seven today. Autodesk has joined the elite world of public companies, placing it with Lotus, Ashton-Tate, Software Publishing, and Micro-Pro in the top 5 visible players in the microcomputer software industry as seen by the financial world. (And don't discount the value of this: a public company gets press coverage as a matter of course that a private company can't buy at any price. Also, public companies are perceived as more solid citizens with more staying power in a competitive situation.)

We agreed to do it right and do it fast. We did both. Name the companies which have moved from start up to public companies in three years. That's a pretty select list to start with. Now look at the ones who have done it with no venture capital, with the original founders still in control, and with not even an outside director at the time of the public offering. You're down to a pretty damn short list. Now filter for the companies with their principles intact: who still believe and practice consistently rewarding the people who do the work, of getting the best people and cutting them in on the pie in a real sense.

So what we've pulled off here is, if not unique, awfully rare in the contemporary business world. This is a good time to reflect on what we've done and to look at how we can best apply the techniques that got us this far to the difficult task in getting to the next plateau.

Mid-Game

Because it's not over, folks! The process of building a company and reaping the rewards of our collective efforts is something I look at more and more as an ongoing brutal winnowing process. Three quarters of all start-up companies fail within the first two years. Only about one in ten thousand companies reaches the stage of making a public offering. And, yes, most public companies languish in the ranks of the NASDAQ Bid & Ask tables where we've taken up residence, rather than becoming the shooting stars who are perceived as the movers and shakers of the industry.

There's a phrase politicians use that I detest. I translate it as "we don't have any idea what the hell to do about this, and things aren't going to get any better". The phrase is "redouble our efforts".[206] I think that this phrase is only used by people who have never in their lives ever doubled their efforts in the first place. I assume that everybody in the founding group of this company is currently working flat-out. What we need to do is continue this, at a sustainable pace, through the next stage in the company's development.

The public offering purchased effective immortality for the company. We now have over ten million dollars in the bank. This means that if our sales went to zero, we could survive for over a year, without any cut-backs or layoffs at our present expenditure level. Given the retrenchments we would make in should such a dire and unlikely (though certainly not unprecedented) scenario eventuate, we could cut back to the core group and spend five or ten years figuring out what to do next (and defending ourselves against shareholder suits). Hell, we could pay a reasonable group reasonable salaries just from the interest on the cash we raised in the offering.

To put the company's liquidity into perspective, I'm sure you remember when we all ponied up our $1 per share to buy Autodesk stock, accompanied by my incessant bleating about how we could run it to the moon. Well, adjusting for the two stock splits we've done, that dollar per share works out to six and two thirds cents per current share, $0.06666. Now today, the company has on the order of $2 *in cash* for each of those shares. So if we all went home and divvied up the pie tomorrow, we would have a gain of over 3000% on our initial investment, or 1000% per year. Not too shabby.

Please bear with me while I do a reverse presentation of these numbers. While the way Wall Street adjusts for stock splits is absolutely correct, people who buy an asset become attached to the price they paid for it (and much investor psychology derives from this). So rather than adjusting the historical numbers for splits, let's look at our performance assuming we never split the stock.

All right, on April 29, 1982 we sold some stock for $1 per share. We issued some options to people who

[206] "Fanaticism consists in redoubling your efforts when you have forgotten your goal"—Santayana.

contributed at $1 per share in May '83. By the time we got to the next round of options, it was all we could do to beat the price down to $2.70 per share in November 1983. Then things really started to cook. The next time we had to name a price was August 1984, when we had to move it to $7.50 (and remember that during this entire period we were doing everything we could to justify as low a price as possible, so that we could issue options as worthwhile as possible to the recipients). The next time we played "pin the number on the stock" was April 1985, and by then it had jumped to $10.50. There's nothing like a public offering to move the stock price to the "industry multiple", and ours sure did. The public offering sold on June 28, 1985 for a price of $165 per share! And last week the stock traded as high as $210 per share. Doesn't that seem different from the quotes you see in the paper?

Does anybody wish he had bought less?

You know, it's really fun writing some self-congratulatory prose after so many "crisis letters" and exhortations to exertion. Just so it doesn't become a habit....

Burning Questions Of Our Movement

So, what happens next? What should we, as the founders of this company and owners of the largest piece of it, be doing to maximise the value of what we've built? What should our company be doing to advance within the industry? How can we best apply the principles upon which we built this company to the very different circumstances and environment in which we now operate? Obviously we've done a lot of right things, but what have we done wrong? What significant opportunities are we, at this very moment, overlooking? And why are our sales only $2 million per month and not, say, ten or twenty million? Can we get there? How?

One fatal luxury of success is a failure to question one's assumptions. We must constantly be looking at what we're doing and the general environment and watch for indications we should be changing our strategy. There is, to my mind, a growing spirit of "we're number 1", "we're unbeatable", and "all the competition is garbage". *This can destroy us!* We have to maintain good morale and believe in what we do, but we have to remember that we got where we are by running scared. There is no shortage of competitors out there with a lean and hungry look. We should be continually reviewing their products and strategies and taking the best ideas for incorporation into our own.

> "The only function of economic forecasting is to make astrology look respectable."
>
> Ezra Solomon

We've talked a lot informally about just what is involved in being a public company. I'd like to put it on paper, just so everybody has the same information all at the same time. Once you become a public company you operate in a fishbowl. Not only is the value of your company and therefore your performance rated daily in the open market, many business decisions you were free to make in private now become open for the world to see. This can lead to making decisions which may be bad for the long term future of the company in order to prevent a cataclysm in the market for the company's stock.

Those who hold and trade the stock obtain information about it primarily from the quarterly and annual reports the company files, from press releases the company issues when important events occur, through reports by financial analysts who follow the stock and are in regular contact with management, and to a lesser extent from presentations the management makes at various financial conferences. Our stock is held largely in institutional

hands. This means that it is mostly in the accounts of pension funds, pooled investment accounts run by banks, and in mutual funds specialising in high-tech. The money managers who run these funds are accountable to the people who put the money in them, and their results are evaluated on a quarterly basis. If a fund is significantly underperforming the market, the money can evaporate as fast as the morning dew on the surface of Mercury. In fact, if a pension fund is underperforming the market, the custodians of it can be personally sued for malfeasance of their fiduciary responsibilities under ERISA. So to put it lightly, these money managers are under a lot of pressure.

They, in turn, look at the quarterly results issued by the company as the major indicator of the company's progress. It's a gross oversimplification, but worthwhile nonetheless to consider the stock price as made up of two components, the earnings (usually expressed as earnings per share or EPS), and the price/earnings ratio or PE. Thus:

$$Price = EPS \times PE$$

The reason for breaking things down this way, is that similar stocks, such as banks, auto companies, aerospace companies, copper mines, and CAD companies will, in the absence of outstanding information peculiar to an individual company, trade at about the same P/E ratios. Thus one talks about the "market multiple" of a given industry. The P/E band moves up and down constantly; in an ebullient market such as 1983, P/E's overall may be twenty times those of a gloom and doom period such as 1974. Autodesk is in a somewhat strange position in that if it is considered a microcomputer software company it will probably settle at a P/E about half that it would command if seen as a CAD company. And of course next year, if software is in and CAD is out, the numbers may reverse. But in the minds of those looking at the stock on a daily basis, the P/E is relatively constant.

Thus, the primary determinant of the price is the earnings per share. This is very simple to calculate: you take our profits after taxes and divide by the number of shares outstanding. Zooming in a bit more, and assuming the number of shares as a constant, our earnings are broken down as follows:

$$\begin{aligned} PreTaxEarnings &= Sales - Expenses \\ AfterTaxEarnings &= PreTaxEarnings \times (1 - TaxRate) \\ GrossMargin &= PreTaxEarnings/Sales \\ AfterTaxMargin &= AfterTaxEarnings/Sales \end{aligned}$$

Now let's look at these numbers and what they mean in the minds of investors. The two key numbers everybody's trying to guess are $Sales$ and EPS. Thus, if you overhear me talking to an investor trying to probe us for information, you might hear me say "we're sticking with 90 cents on 25", which translates to "Look, I hope we really blow the top off the industry and end up with the whole pie, but I sure don't want to be dumped on if 'all' we do is increase our sales by 250% this year. We're 95% confident that, assuming no changes in the current competitive environment and the economy as a whole, that our sales will be at least $25 million and we'll earn at least 90 cents per share of outstanding stock."

As each set of quarterly results are issued, they will be eagerly digested for indications as to whether the company is ahead, on, or behind expectations. Investors want to see each quarter increase both $Sales$ and EPS from the last, and compare each quarter's results with those of the comparable quarter in the previous year to see if growth over the year matches the expected growth rate. To date, our business has not been seasonal, so straight quarter-to-quarter growth will be expected.

Before we move on to the edgy relationship between the company and the financial analysts who cover it, I'd like to define "visibility", a key term in that relationship. Visibility measures to what extent outsiders can

predict the business trends of a company overall. Consider a defence contractor. In that business, you receive contracts to do work, and the size of the contract and the payment terms are specified in advance. Any changes in the contract are public documents and are disclosed immediately in any case. Thus income is calculable by anybody who reads the paper. Expenses tend to also be pretty well predictable from historical measures, so all you need to come up with pretty reliable sales and earnings forecasts is a subscription to Aviation Leak and a pocket calculator. This is a business with high visibility.

Now let's consider a hypothetical company whose sales are almost entirely booked over the telephone. Most orders are shipped within 48 hours of receipt, so there is no backlog and no sales contracts to forecast. If the phone stops ringing, the money stops flowing. This company's sales flow through many different kinds of outlets and into numerous markets, which may behave differently as economic conditions change. The product costs almost nothing to manufacture, and is sold for a high price which is justified by difficult to measure productivity measures. The high price is largely the result of a lack of competition in the market; a determined competitor could sell such a product for $100 and make money doing it. The expenses of this business are mostly sales and marketing expenses, which are determined by the need to respond to competition and open new markets. Such a business would have really lousy visibility. I leave to you the exercise of naming such a company.

So who do those whose jobs are on the line turn to in order to decide if they should buy or sell our stock? The security analysts. These analysts usually work for the various investment bankers, and "follow" a group of stocks, usually in one industry. The analysts initially following our stock are Peter Schleider of L.F. Rothschild and John Rohal of Alex. Brown. We hope additional analysts follow our stock in the future. The analysts write regular research reports on the stock, and talk to management in order to prepare their own estimates of the company's future. These reports are then used by the institutional sales forces of the bankers to sell stock in the aftermarket. An analyst will probe to get as much information as possible, and then issue his own forecast. In some cases this forecast may be much more optimistic than that issued by the company. If the company fails to meet the forecast, the analyst will then write a report which says that the company "had disappointing earnings", even if they represented a new high and exceeded the company's own expectations. Now this may be a little hyperbolic, but it has happened, and it does happen. Maintaining close contact with the analysts and seeing that they reach the conclusions you want is an ongoing task for a public company.

In the offering process we "signed up" to a set of performance criteria. Our investors will be watching these and, having been sold very many high tech stocks that went south soon after the offering, will be using them as triggers to dump the stock. *We cannot let this happen*. Therefore, here are the numbers by which we live and die. All of these numbers are consolidated, i.e., the sum of domestic operations and all foreign subsidiaries. We must do $25 million in sales this year (FY ending January 31, 1986). We must generate 90 cents per share after tax profit. Our gross margins must be in the band from 35% to 40% and therefore our after tax margins should be about 20%. We must build the company and our distribution channels and product line to support $45 million in sales the following year and $1.45 per share after tax profit.

These are the company's must-meet goals. We hope to do a lot better, but we *must not* do worse. If we fail, the management will be battered by the shareholders, and our stock will be gored. But the management cannot make these goals happen. The company as a whole must do this. I tried to involve as many as people as possible in formulating these goals. Now we have our job to do. Let's get on with it.

What will be the environment in which Autodesk will be operating in the future? First of all, we cannot spend the proceeds of the offering on virtually any of the needs we perceive the company to face. Since in the software business we don't use any expensive capital equipment, virtually everything we would spend the money on shows up as an expense on the income statement. If we hire people, that's salary. If we do an advertising blitz,

that's promotion. From the standpoint of accounting, spending the money we raised in the offering is precisely the same as spending money we get from selling an AutoCAD. Now please refer back to the equations given above which calculate the critical numbers. If we spend the money from the offering to hire people, or to advertise, or to do any of the obvious things, those dollars are added to Expenses. That gets subtracted from Sales and reduces earnings and margins. Assuming a marginal tax rate of 50%, the reduction of the pretax numbers is twice that of that of the after tax numbers. And remember that our performance is being watched *quarterly*. Even if we can spend the money knowing that it will generate a major return in six months, that's not good enough. The added expenditures will affect the one or two intervening quarters, and Autodesk will be perceived as having "disappointing earnings" or, even worse, "eroding margins" and look out below. (If you think for a minute you'll see why eroding margins are a superb leading indicator of competitive pressure.)

Now this may seem to be a lot to digest, but it really is crucial to the way in which the company will continue to operate. When I say, "We have all this money but we can't spend it", I am not setting up a smokescreen to deny people in the company what they want. I'm just describing the reality which I hope the above has somewhat clarified.

What Business Is the Company In?

This is a computer software company, y'hear. Maybe this is sufficiently obvious that it doesn't need restating, but as the company grows there is a tendency for every department to look upon what it does as central to the mission of the company. Departments then tend to see if they can make a direct contribution to the till by adding products to the company's line. For example, training could offer courses to users around the country for a fee. Technical could offer consulting services to driver developers for an hourly charge. QA could perform screening of third party software products for vendors. Marketing could prepare promotional materials for OEMs and third party vendors. And Production could manufacture third party products. Within the next year, we may be doing any or all of these functions, and these activities may be contributing dollars to the company's revenue totals. *But they are incidental to the business the company is in, which is designing, developing, manufacturing, selling, and supporting computer software!* Our company's value largely derives from the fact that what we do is so extraordinarily profitable. It is so profitable because we are selling intellectual property; virtually pure value added; pure reason without the critique. As I said three and a half years ago in the original Working Paper, there are few legal businesses as attractive. If we wish to go into another business, we must review that proposal as we would review the purchase of an operating company: looking at capital requirements, sales projections, pro forma income statements and balance sheets, and risk factors. This is one of the most profound decisions a company can make, and is not to be taken lightly or backed into inadvertently.

If you aren't used to thinking in the terms expounded above, the impact of stumbling into a new business can be less than obvious. Suppose we were to start doing direct contract support to major user accounts. Suppose that this was so extraordinarily successful that by the end of the year we had generated 5 million dollars in support fees, and had managed to do this with expenses of 4 million for personnel and travel. We would then add 1 million dollars to the pretax profit number and $500,000 to after tax profit. Sounds great, right? Wrong. That component of the business would be operating at a 20% pretax margin and a 10% after tax margin. When these numbers were consolidated with software operations, they would reduce our operating margins, and Autodesk would be perceived as having eroding margins. The analysts would then look at the numbers to find out what was happening and discover that we had gone into the education business. Education is not a stunningly profitable business (as thousands of colleges know, and Westinghouse and CDC learned to their dismay), and Autodesk would lose some of the attractiveness of being a "pure high-tech CAD play". This

would reduce our P/E, and the stock could be clobbered.

I don't want to dwell on this too much, but it is a problem that growing companies typically have. We'll have to keep focused on the ultimate goal of selling a lot of software if we're to avoid it.

Product Style

I'd like to talk for a moment about our products and their general style. I've spent some time recently using other people's software packages and fooling around with some new product ideas, and it's clear that many of the things we talked about happening three years ago have happened. The micro software business has become very professional very rapidly. The standards for user interfaces and ease of use have risen extremely rapidly. I'd like to talk a bit about some of the implications of this.

Why do we make clunky user interfaces? I think that some of our much-vaunted "mainframe approach" to software may be leading us into some poor decisions in the current environment. We always build software to be easily ported, machine independent, and easy to maintain and enhance. These are things much to be desired and unqualifiedly good, *as long as there is no cost to the user*. If ease of development or support imposes a performance, convenience, or learning cost upon the customer, this must be looked at as a tradeoff, not decided preemptively in favour of the developer.

I would invite you to spend 5 hours using a program Dan Drake turned me on to, "Managing Your Money" by MECA. This is a $120 program of extraordinary complexity. I would rate its connectivity and integration as approaching AutoCAD. It fills three discs. You have to read about three pages of manual to get started on it, and it contains hundreds of pages of intelligent, useful, and witty on-line assistance. All response is absolutely instantaneous. No error is fatal. You can always back up.

I think our tendency is to adhere to the reggae rule of "all killer, no filler" in designing our programs. We tend to eschew user interface "fireworks" such as instantaneous screen updates, fill-in-the-forms data entry, pop-up menus, and function keys in favour of solid, well engineered but prosaic programs.

This is one of the assumptions we should question.

How long are users going to accept a product which requires mastery of a 300 page manual? In a market which is dominated by IBM and compatibles, what is the opportunity cost of not deriving greater advantages by tailoring to these machines? Should we do a Macintosh or Atari product? I don't have the answers to these questions. But I think we have to consider them.

What do we do next?

On the operational side, I think we should pledge a significant effort to doing this job of being a public company right. We showed the Silicon Valley cynics and the venture capitalists how a bunch of dedicated, talented people could do a start-up company right. Now let's show Wall Street how a small public company should be done. What I propose is that (all within the limits of the law, of course) we treat our stock as an Autodesk product. Let's put together a shareholder communication program that rivals the ones we've created for our dealers and users. Let's put on shareholder forums at our annual meeting just as we do for the dealers and conference

speakers. And let's have an informal shareholder forum at the 6 month point between meetings. Shareholders can be a pain in the ass, but they do own the company. And let's see what we can do to make this perceived as a very special company to own stock in. We must be open and candid. We must present realistic numbers and always meet them. If we're not going to meet our numbers, we must give warning as soon as we know, and have explanations ready to deliver. These acts will build loyalty that will stand us in good stead when times get tough.

The Next AutoCAD

> "There's absolutely no way we could put a man on the moon by 1994. No way."
>
> NASA Official, quoted by Jerry Pournelle at the 1984 L-5 Convention

It's also time to look for some new product ideas. Let's not settle down into this "going concern" mentality where we're maintaining a program that came from "somewhere" and is going "somewhere". We built this from zero. We can do it again. Can a company our size continue to develop multiple products on the cheap, test market them, and get behind the winners? Can we rekindle the old "ten wolverines in a barrel" technological ferment we had when the idea of this company was fresh and new? Can we have more yelling and less nodding? Will *somebody* please come to me and ask to spend a week in another department to learn what the hell they do in there anyway? What is the minimum time in which we can bring a product to market now? Could we bring a new product to market this COMDEX? Look at the development logs to see where AutoCAD was at this point in 1982.

The American Dream

> "Conrad Hilton didn't make his fortune by building only one hotel and then opening a car wash. He may have had a car wash before he had a hotel, but once he found hotels and they succeeded, it was hotels, hotels, and more hotels. Did Ray Kroc who owned McDonald's start one hamburger stand and then open a dry cleaners? No. He had a winner, he stayed with it, and wealth was accumulated."
>
> Dr. Bruce Gould, "How Fortunes are Made", in *Bruce Gould on Commodities* (newsletter)

There are two ways to look at what we've done here. We're either an awfully lucky bunch of weirdos, or we're really on to something. With every passing month and every milestone we reach, the luck seems less likely and the strategy seems more astute.

Let's franchise it.

Franchise it? Well, not exactly, but here's the idea. We started out as a group of people with limited financial resources but a great pool of diverse talent and willingness to work hard. We built this company as opposed to going to work somewhere else because we felt that this was the best way to achieve the success we wanted to reward our exertions.

I don't think we were the only people in the world with these goals. Let's build ten, twenty, or fifty more Autodesks. How? By offering the same kind of partnership to entrepreneurial people that Marinchip offered to the founders of Autodesk. We publicise the following proposition:

You want to start a company? We know how to do it. Look at our results. Put together your group, count your money. Here's a set of information letters that tell you how to do it. Come to us with your proposal. We don't want a long business plan; you don't know what's going to happen any more than we did, and if you say you do you're a bullshit artist and aren't worth listening to.

We'll look at the people and the product ideas. Is there that sparkle you can see in the first 5 minutes? If we're believers, we'll match your investment dollar for dollar. In return we get 35% of the company, held as soft preferred which basically protects us against being ripped off, but we're in there with all the other founders. But it's the founders' company. Our investment in these companies will not be an expense on our balance sheet. The investment will purchase stock which will be carried as an asset. I'm not sure how often such illiquid assets get marked to market, but we can probably let it just sit there until we either write it off or begin to get proud of it.

Autodesk will provide limited support to the venture. We will see that the legal details are taken care of correctly and that the accounting is of public company quality. We'll provide a pool of talent to the management that has "done it before". We'll offer technical resources and the facilities of our lab on a sporadic basis. Our distribution channels, marketing and promotion resources, and our ability to promote products at shows and in publications at small marginal costs will be important resources to companies affiliated with us. Our manufacturing and shipping operations can provide those services at low costs per unit.

The founders of the new companies can choose to use our services, which will be billed at attractive rates, or to build their own or go outside—it's their company, career, and destiny. We will be providing what all the venture capitalists claimed they did, "bringing more to the table than just money". The only difference is that we really will.

Our goal is to give the founders of these new companies the same shot at success we had when we started. Autodesk will provide some cash that we sorely could have used, but not enough to mess things up, and the ongoing establishment the creation of which cost us so many critical hours we could have better spent elsewhere. We won't run the show, but we'll try to be there when we're needed. Many of these companies will probably fail, but if 20% work, they will contribute mightily to Autodesk's success.

Just imagine if we pull this off. I hope we always retain some of the rabble-rouser elements of our creation. I can think of nothing I'd like to do more than drain the talent out of these Silicon Valley companies that are screwing their key people and giving the equity to the venture capitalists. Instead, here will be Autodesk, with one face talking to security analysts and breaking new ground as the model for small public companies, and at the same time erecting a rickety, low-rent conglomerate built on talent and hard work, of hardscrabble start-up maniacs who, just once, want to do something right and own it.

I think that people would be well served to take the chance we'd be offering. They'll have to have *real* commitment, *real* performance, *real* responsibility, and *real* professionalism to make it. If they're interested in making that kind of commitment, we can't guarantee that they'll succeed, but we can guarantee that together we'll have a once in a lifetime experience as we try. (Working paper, 1/82, Page 24).

Now putting this together will take some work. But how much, really? Let's think about it, and see if we can pull another sleeping shocker on the industry. Can you imagine, just imagine, ten companies, all loosely affiliated, working like Autodesk all at the same time. Why they'll say it's a movement.

And that's exactly what it will be.

Time of Turbulence

After a period of great strain and tension within the company, Mike Ford, who was, more than any other person, responsible for developing the marketing and sales strategies that resulted in AutoCAD's enormous success, and in building the marketing, sales, and support organisation from a single person (himself), to more than 80 people, submitted his resignation.

This is the memo I issued announcing his departure and the text of the remarks I made at the company meeting the following day. This was not a happy time for anybody in the company, but it was a significant event in the company's history, so these documents are included here.

Memo to everybody

To: Everybody
From: John Walker
Date: 5 February 1986

Mike Ford has submitted his resignation as Vice President of Marketing and Sales, effective today.

I have accepted his resignation.

Effective immediately, Richard Handyside will assume the role of acting Director of Marketing and Sales. Richard continues to hold the position of Vice President of European Operations, and will be spending one week per month in the London office to continue his work there. Richard is a founder of Autodesk, Inc. and has been responsible for building our sales and marketing efforts in the United Kingdom since the company's inception. Please extend him the help he will need in filling the job he'll be assuming.

We will immediately begin the process of finding a permanent replacement for Mike Ford.

The events of the last two weeks, and indeed the last several months have placed many of us under extreme stress. We have all found ourselves wondering what was happening as rumours circulated and we considered what they might mean for the future of our company. We have come this far by trying to build a company which was open, fair, and honest. That is how we must continue.

There will be an all-company meeting tomorrow in the administration area (where we usually have the monthly meeting). Because of the extraordinary seriousness of what has happened, the meeting will be longer than usual, and will start promptly at 5:15 P.M. I urge all of you who are concerned with the future of our company,

unsure as to what has been happening, upset with the way things have been handled, or just confused to attend. I will try to explain what has happened and what happens next. I will answer all of your questions. I will stay until there are no more questions. All of the rest of the management of the company will be available to respond to your concerns.

In addition, I would like to reiterate that this company has always had, and will continue to have an open-door policy. If there is any matter you want to discuss with any of the management of the company, please bring it to us directly. If there's anything you'd prefer to discuss on a one-to-one basis, that offer stands and will stand.

If we continue to build our company on the principles which got us here: honesty, hard work, rewarding the people who do the work, and striving to minimise the politics, we can look back on this period as a time of testing for the company as it continued to grow. We have had what it took to build one of the singular successes of the 1980's in only three years. We have what it takes to continue. Let us begin today.

Remarks at the company meeting

Thank you all for coming.

This has been a time of great stress on all of us. Our energies have been diverted increasingly from the tasks we all need to do to take advantage of this tremendous opportunity we share and spent on speculation about people, about events, their consequences, and the future of our company.

Those of us who have been here since the company started in my living room on January 30, 1982 know that we tried to build something very special here. We were all people who had worked for companies large and small and always saw those companies squandering much of their resources on politics, on empire building, on image, on status symbols, on so many things not connected with success.

Like fools perhaps we imagined a company that worked differently, a company that concentrated entirely on the clearly defined goal of success in its industry, producing products of the highest quality, marketing them honestly to people, and efficiently delivering the fruits of our labours to the people who made it happen, rather than to a bunch of nameless investors or managers.

I won't recount all the times people said it wouldn't work; couldn't work. That it would all come apart when we reached 25 people, or 50 people, or 100 people. That when our sales reached $100,000 a month, or $1 million a month and we had "a real company there" we would be like everybody else, that the magic would be replaced by plodding, the creativity by repetition, the innovation by mediocrity. That is, after all, what we'd seen happen so many times before in so many other companies; it was only really *fun* in the early days when you were losing money and everybody was struggling together to survive, doing any job that needed to be done, working whatever hours it took, expanding their scope of expertise as the company called on them to grow, and grow fast into new jobs.

And then the money starts to come in, and the walls go up, and the hierarchy begins to grow, and the restrictions come down, and before long it's the "job" and not the "challenge". And sometimes when that doesn't happen, we read about companies that just come apart because of too rapid growth.

Is that happening here?

I think not.

There has been some changes. We have lost some people who have been here from the very early days. We have lost people who have worked long hours, applied great skills, and whose dedication to building this company, and unique insights into strategy and the way to make it into reality were major contributors to our success. We will sorely miss these people.

But let us look at what has not changed.

Our company is the acknowledged leader in its industry.

Our main product, AutoCAD has a market share in excess of 50%. This is a situation that only rarely ever occurs in industry, and presents the company that achieves it with virtually unlimited opportunities.

Our two new products, AE/CADD and CAD/camera have, in less than 6 months, both emerged as the unit volume leaders in their respective markets. We are making money on both products. We are on our way to success with both.

We dominate the channels of distribution for our products. Almost all of the dealers qualified to sell CAD sell AutoCAD. We have OEM arrangements with a list of companies which represent a Who's Who of high technology. Our position in education is commanding, and our innovation in addressing that market has been so great that even those who try to copy our strategy can't keep up.

Out oft-repeated goal was to make AutoCAD the standard for CAD on this planet. Today AutoCAD *is* the standard for CAD on this planet.

We have built a vital and fast-growing applications program. By putting tools in the hands of creative people, we've brought about an efflorescence of creative products that other CAD companies have failed to produce given decades. And we've done it in less than two years.

Our marketing effort has consistently won awards indicating it is the best in the industry. It is almost unheard of for a company our size to go it alone, spurning ad agencies and PR firms. Our team in Marketing has worked miracles with meager resources. The people who did that work will be given more resources to continue it. And I am confident that this group of people, who have shown themselves to be the best in the industry, will continue to distinguish themselves with continued awards and with results.

Autodesk isn't about safety and caution. It's about going for opportunities and conceding nothing to the competition. We've set up Project Gold, to address the Fortune 1000. It's a program unlike that of any other company. That's never stopped us before. And unlike so many other programs, ours is going to work. It's already beginning to work. And it will contribute mightily to our results that we gather to discuss this time next year.

We're also going after the government market for CAD. That's a very different kind of selling job than the one we've been doing so far. So what? We'll make it work too. And the goals here are high.

In the operation of the company that generates the money, we've grown with remarkably few pains, and we're running well. We're taking more orders from more dealers, processing them efficiently, manufacturing them rapidly and correctly, checking them so that we put the best product we can manage in the customer's hand.

Our support has been praised in *InfoWorld*. It is repeatedly praised in letters we receive. Few companies have even tried to support a product as complicated as ours over the telephone. None has done it as well.

Our accounting and financial operations continue to be a model of smooth operation. And the phones are

working better.

Our development of the next AutoCAD continues, aiming for entry to Beta testing this month. This new product will have an impact as people begin to realise what they can do with it as major as the introduction of AutoCAD in 1982, or of AutoCAD 2.0 in 1984. Our competitors simply have no idea what is about to happen to them. Their products, placed against our new AutoCAD, will be bows and arrows against the lightning.[207] We're wading into the workstation market with Sun, Apollo, MicroVAX, and IBM PC RT products in the mill. All those high priced players now have to contend with AutoCAD and face the fact that AutoCAD is what the buyers want.

Our new products continue to progress. We've moved AutoLISP from concept to a shipping product. We have two new Architectural products under way, as well as a major upgrade to AE/CADD.

And there's a few wild ideas in the back room that are going to turn into industry-stunners before all that long.

That's what hasn't changed. We're a smoothly running, lean and efficient, creative, productive company. We've just completed our fiscal year at the end of January. We came in on target, with sales and earnings exactly what we aimed for. Once again, we've met our numbers. Every time we do that, we convince more people that we're going to be around for the long haul.

Last month, I got up here and said we'd do something about profit sharing. We're doing it. We've taken some of our profits and we're giving it to all of you who made it happen. Unlike some companies where the management skims the pot, we're doing it like this: everybody who's worked here full time for the full year gets $1000 of bonus (we have to do tax withholding on this, of course). If you've been here less than a year, or work part time, you'll get the proportionate amount for the time you've worked. That's it. It doesn't matter what you get paid, what you do, or who you work for—you get $1000. You deserve it. You earned it. Thank you.

Let me promise that we'll continue to do something about profit sharing in the future. I'll share the details when they're firm and I can make a commitment I know we can keep.

So let's talk about what has changed. As I said, we have lost some people. I don't want to go into details here of what happened when, who said what to whom, or what who said what where. Events and situations occurred which led me, in conjunction with all of the current senior management of the company and most of the founders to conclude that we were going down the wrong road with the company. We decided to remedy that situation. We decided to eliminate the stress which was hurting us all, and to pay the price of the consequences. We made our decision and we acted. Now things will get better if we all work to make that happen.

The politics stops today. The whispering stops today.

The honesty, openness, and fair dealing that built this company is back. We never intended it to go away, but maybe we were so preoccupied with immediate problems that we failed to reaffirm what has made this company so different and so successful. If you have any questions about what has happened, or have any concerns about what is going on, you can discuss them with me or anybody else in the company regardless of position, in private or in groups.

This company is not supposed to be run top down. Our success has been a consequence of how well we listened to our users, our dealers, our OEMs, and others we deal with. Information has to flow from the people in direct daily contact with these people *up*, so that the right decisions get made where to apply our limited resources.

[207] This was what was released as AutoCAD 2.5. Among other things, it introduced the hardware lock to the domestic market.

We will not falter because we lose some people at the top. We will not falter if we were to lose everybody at the top, as long as the people who are really doing the work continue to do it, and continue to listen to what they hear, share it with others in the company, and act to meet the needs they feel.

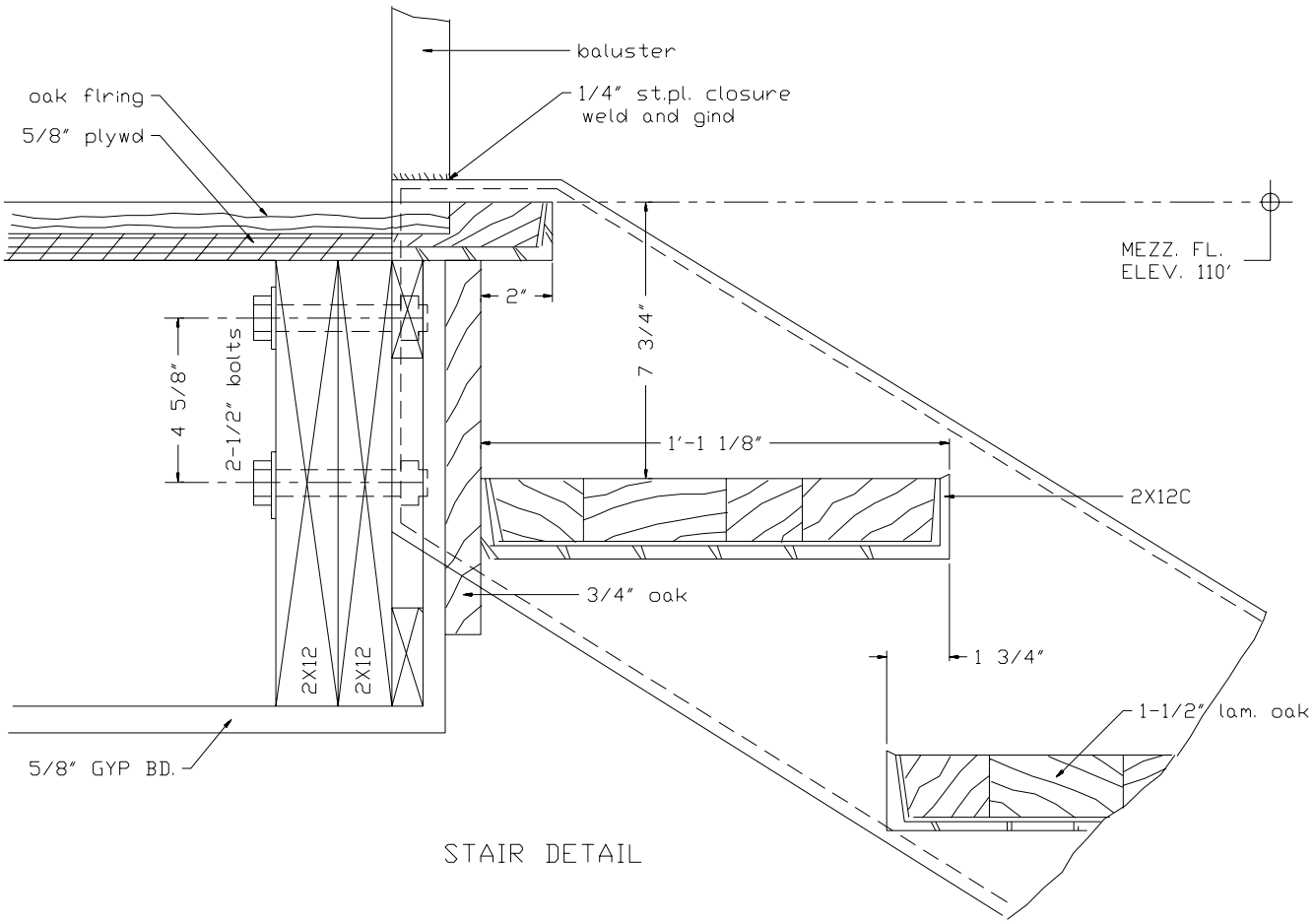
I don't care what you do in this company. If you think we're heading in the wrong direction, if you smell a problem we're overlooking, if you see a threat we seem to be ignoring, tell me or somebody else in the management. This isn't a privilege. It's part of your job and key to our long term survival.

Just a week over four years ago, 16 people decided to start a company and do it right this time. Today, 160 people have the opportunity to start from that base and build something much bigger, so that four years from now we'll all be the envy of the latecomers, the founders who were there in the early days, the people who saw the opportunity become reality.

Mankind is the animal that makes tools. In each generation, only a few people get a chance to create new tools. Very, very few get to contribute to making a tool that changes the lives of first hundreds, then thousands, then tens of thousands, and someday millions of people. We are in that position. Our work so far has put us there.

I feel privileged to have shared this experience with all of you. Now we keep on working together to pursue this opportunity. Few ever get this kind of a chance. Rarely is there such an opportunity to so immediately see the consequences of your hard work. This is the fulcrum, the point where we make the potential of AutoCAD really begin to change things in a large way, and where we decide to keep this company on the track that brought us so far so fast.

Peter Barnett drew this architectural stair detail in 1984 to illustrate architectural applications of AutoCAD. It has appeared in numerous advertisements and brochures, and has been on the sample drawings disc from AutoCAD version 2.0 to date. Bob Elman spent a great deal of time cleaning up this drawing into its present form.



STAIR DETAIL

Speech at Silverado

Every year, InfoCorp (an industry analysis company) holds a technology forum at the Silverado resort in Napa, California. I was foolish enough to volunteer to speak, and so I had to prepare this speech which I gave with slides. I'm including this for two reasons. First, this is the most coherent exposition in print of Autodesk's strategy at the time. Second, preparing and delivering this speech, on top of everything else that happened in early 1986, was one of the items that prompted me to start seriously planning to hand the job of president on to somebody better suited to do it.

InfoCorp Silverado Speech

by John Walker
Revision 2 — March 2, 1986

The machine age began to come to an end with the invention of the first programmable computer.

We often forget that the word "technology" comes from the same root as "technique".[208] It has nothing to do with machinery, it's how humans apply their minds to solving problems.

The general purpose computer is a tool which allows pure technique in the form of algorithms to be applied to problem solving. This is the central fact of the computer revolution.

Since the advent of the low cost microprocessor, we have seen the replacement, on an accelerating basis, of special purpose machinery (whether mechanical or electronic) with general purpose computing elements. Examination of a 1970 vintage teletype beside a contemporary printing computer terminal will illustrate the extent of this revolution.

Today, as the designers of the complex machinery of the past retire, the skills which created such exquisite machines as mechanical calculators which could divide are moving from practice to history. Designers entering the workforce often view the achievements of their seniors, accomplished without computing elements, the way our civilisation views the building of the pyramids.

So what in the world does any of this stuff have to do with PC's in the engineering world, or with PC CAD?

In every area general purpose computers have entered, they have been forced to educate people that what was once a machine is now simply a piece of software.

[208] The Greek $\tau\epsilon\chi$.

•••

Remember when computers looked like computers?

This is the first computer I ever used.[209] By the standards of the early 1960's, it was a supercomputer.

It had 256 thousand bytes of 8 microsecond core memory.

Its magnetic drums provided 6 megabytes of secondary storage.

It added two 36 bit numbers in 4 microseconds.

It performed single precision floating point adds in hardware in 14 microseconds.

It communicated with remote terminals at 2400 bits per second.

It served a single user at a time, with high speed batch processing.

And laser technology was on the verge of revolutionising data storage.

This computer performed all the engineering, scientific, and software development computation for a university with a graduate and undergraduate population of over 2000. It was used for finite element analysis, fluid dynamics, particle physics, compiler and operating system development: virtually every field of science and engineering. It was retired in 1968.

•••

This is the computer I'm using today—an IBM PC/AT, the exemplar of the PC in technical applications.

It has 3 and three quarter megabytes of 150 nanosecond RAM.

It has twenty megabytes of disc storage.

It does 32 bit adds in 2.3 microseconds.

It provides both single and double precision floating point in hardware, doing single floating adds in 20 microseconds.

It communicates with other computers on a network at ten million bits per second.

And laser technology is on the verge of revolutionising data storage.

It serves a single user at a time, me. It spends more than half of its time turned off, and a majority of its time while on waiting for me to type on the keyboard.

This computer, a PC, has 14 times the main memory, and three times the disc storage of the 1960's mainframe. Its memory is 53 times faster. The processor is 3.4 times faster for integer calculations, and .7 as fast for floating point. For double precision floating point, the heart and soul of scientific and engineering work, the PC is over 4 times faster.

[209] The Univac 1107 at Case Institute of Technology, Cleveland Ohio.

So it isn't just hyperbole when we talk about having a room sized mainframe's power on our desks. For less than the price of a car, we can own a computer more powerful in every way than the mainframes on which most of the key engineering applications used today were written.

• • •

It is these statistics which show how utterly absurd it is when somebody pronounces that some job or other will never be done on a PC.

• • •

The confluence of these two trends; the displacement of special purpose machines with general purpose processors, and the ongoing giddy decline in the price one must pay for computing power has led to the development of the general purpose workstation.

In the mid 1970's, everybody thought of a word processor as a machine. It was an expensive box, bought from a word processing company primarily by large corporations, who could pay the price to obtain the productivity gains such a machine delivered. Virtually beneath the noses of the word processing vendors, people started selling word processing programs which ran on PCs, those funny "hobby computers" that strange people bought and played with. Before long, the use of general purpose PCs with word processing software dwarfed dedicated word processors, even in the markets where word processors were strong. Note that they did not supplant word processors—companies continue to buy word processors for full-time typists, and word processors provide services to these operators which PC programs do not currently provide. But most people write as part of their job, not full time. They need an easy to learn and use tool which is one of a collection of tools they run on a personal workstation.

In 1982, my company started selling a computer aided drafting and design program which ran on PCs. The conventional wisdom, as represented by those venture capitalists and analysts we could get to talk to us was:

1. You can't do CAD on a PC.
2. Even if you could, no serious user would buy it.

and 3. Computer dealers can't sell CAD systems.

Well, we didn't have anything else to do, so we just went ahead and tried anyway. To date, we've sold in excess of forty thousand CAD packages for PCs. To put this number in perspective, it is on the order of twice the number of workstations of the most widely used mainframe CAD system.

Nobody can afford to discount PC CAD and PC engineering applications today. Let me give you some statistics about AutoCAD, the program I sell. I'm using our program as an example because I know the numbers for it; there are other PC-based engineering applications with similar statistics.

The source code for AutoCAD is in excess of 200,000 lines of C.[210] The program is well in excess of a megabyte; programming tricks from the 1960s allow it to run in much smaller machines. Today, it embodies over 70 man-years of development. It is being enhanced at the rate of over 20 man years per elapsed year. It runs on over 30 machines, supports in excess of 120 graphic peripherals, and operates compatibly on MS-DOS, Unix on the Sun and Apollo workstations, AIX on the IBM RT PC, and VMS on the VaxStation II.

[210]Four years later, AutoCAD Release 12 had grown to more than a million lines of C, not counting drivers or applications such as AME and Render.

Over 150 third party application packages have been interfaced to AutoCAD. These include structural analysis, bill of material extraction and job costing, pipe stress, architectural design, numerical controlled machine programming, municipal mapping, surveying, printed circuit autorouting, and even football play diagraming and theatrical lighting design.

AutoCAD may be used as a standalone application package, or it may be programmed for specialised applications. We have integrated the LISP language into AutoCAD, allowing users to extend the system for their own jobs.

Over 600 educational institutions teach drafting and design with AutoCAD. AutoCAD is available in English, French, German, Swedish, Italian, Japanese and Spanish language editions. Translators are available which allow AutoCAD to interchange drawings with most major mainframe CAD systems.

By virtually any measure you choose to apply, lines of code, internal complexity, investment in development, ongoing development commitment, open architecture, third party support, migration to multiple hardware platforms and operating systems, and computing power consumed by the package, AutoCAD is mainframe software. So why do we choose to sell it on PCs?

Because that's where the money is.

It seems like if you stand on a street corner in Silicon Valley and hand somebody a $20 bill, he runs off to build an engineering workstation. On everybody's mind is the refrain, what will become the engineering workstation of the 80's and 90's? Chip makers vie to position their 32 bit processor as the heart of the lucrative engineering workstation market.

The numbers tell another story. Today, for every 32 bit engineering workstation in the world, there are more than one hundred 16 bit MS-DOS machines. It's news when a major company selects one of the workstation vendors to provide 200 workstations for an engineering facility. It's routine when their purchasing arm orders another thousand PCs to equip another department. They are, after all, just PCs.

As a software supplier, I can't help but notice that those who sell software on the 32 bit workstations don't get a hundred times as much per copy of their software as those of us who sell on PCs get for ours.

Let me return for a moment then, to the theme of this talk... PCs invading the engineering market. I'd like to retitle it at this point... to PCs *infiltrating* the engineering market. I'll explain how this is happening in a moment.

First of all, please keep in mind that only a small percentage of the engineering and scientific work in this country is done in the large companies. I'm fascinated by how many companies focus on the Fortune 1000 to the exclusion of the rest of the market. Marketing to the Fortune 1000 has its advantages; it's a clearly defined prospect list, heavily researched, and generally sellable from the top down. But in many cases it's only a small fraction of the market.

There are over 600,000 manufacturing organisations in this country. Eighty five percent employ ten people or less. The overwhelming percentage of architectural firms employ less than ten people. Even within the largest companies, there are sales offices, project groups, and application engineering arms which are operating as small autonomous entities.

Most drafting and design done today is not being done on CAD systems, it's being done by hand. Most mechanical parts are not being designed on mainframes or engineering workstations, they're being designed on paper, or by rule of thumb. Only a tiny percentage of the entire building process, from architecture through

construction to facility management, involves computers at all. This is the vast untapped market. This is what accounts for the dominance of the PC in engineering applications. Those who ignore it do so at their peril.

An engineer's job embodies many different activities... writing, reading, performing calculations, drawing, interacting with others, and exploring design alternatives. The computer can play a part in all of these parts of the job other than face to face interaction. An ideal engineering workstation, therefore, is one which performs all of these tasks while requiring the least effort to master.

Engineers spend a lot of time writing. They do calculations which in the dark days before microprocessors were done on slide rules. Those in larger organisations access central databases, and may use some form of electronic mail for communication. There is a cheap and effective tool which improves productivity in all of these tasks—the desktop computer. For less than the monthly salary of an engineer, a computer which delivers immediate and measurable gains in output can be placed on his desk. And that's being done today, by the millions.

Don't underestimate the power of a widely distributed tool. If an engineer has a PC on his desk, which may have been purchased, justified, and primarily employed as a word processing station and terminal to the company computer, and has the choice of, for example, doing a drawing on that desktop machine with a PC CAD program, versus signing up for time at the corporate CAD centre, or submitting a sketch to the drafting department for three day turnaround, what will be his decision?

That decision will usually be made based on the engineer's desire to optimise his own productivity. Who cares if doing a drawing on the PC is ten percent slower than using the corporate CAD station... if that ten percent amounts to 6 minutes for a one hour drawing, and you have to wait 2 days for your time on the CAD station, or 3 days for drafting to turn around the drawing, the engineer and his employer is well ahead by using the PC to make the drawing.

Similarly, in the small organisation, the purchase of a PC may have been prompted by the desire to use it as a word processing system, or to keep accounts, or to maintain spreadsheets. The addition of a PC CAD package, or other engineering or scientific analysis package may result in productivity gains in those aspects of the business as well.

In both cases, then, what we're seeing is infiltration, not invasion. The PC is justifiable as a productivity tool in organisations large and small. It is bought to perform the tasks that everybody agrees it does best. Once in place, users discover that they have far more power at their fingertips than they imagined at first, so the PC assumes more and more complex tasks, impinging on those applications oft considered as requiring a "mainframe" or "turnkey system".

The PC, because it is there, is applied to these more demanding tasks. Users discover that the PC does a fine job performing them, and as the word spreads, the managers of the central data processing and CAD facilities discover that the projected growth in demand for their services is falling below expectations. First, by a small amount, then a larger factor, and finally by so much that somebody is delegated to find out what is going on.

And another organisation discovers the PC as engineering work station.

• • •

Every time I give a talk, I always get a question that goes like, "Do you see the 32 bit machines supplanting the 16 bit machines", or "what will the impact of RISC technology be on the workstation market", or "will the

80386 cut into the 68000 based market". I have to answer by saying, "I don't have any idea and I couldn't care less".

Except for those who build the machines and those who invest in them, the issue of CPUs and operating systems is of a lot less significance than a lot of people think.

Let's look at the general shape of a workstation as it will exist in a couple of years. It will have a fast processor, delivering at least the power of a VAX 780 on the desktop. It will support an operating system which provides multitasking, and will probably look a lot like UNIX. I include MS-DOS and its derivatives as Unix-like. It will have a high resolution screen, and the software will support a window environment allowing concurrent execution of disparate tasks. The station will be usable as a standalone computer by a user with no previous experience in system administration, and it will be possible to connect the system to others in a network which provides a true distributed file system.

It will provide all of the programming languages in which major applications are written, including C and FORTRAN compilers of high quality. All of the major engineering and scientific application packages will be available to run on it.

And, sad to say for the vendors, it will probably cost a lot less than many of them assume today. More on this later.

Numerous contenders meet, or will soon meet these tests. The software suppliers will put their packages on all of the major contenders' machines. The market will decide who the survivors will be.

• • •

The terminology in the computer industry has always been murky. The fact that we don't break out in laughter when we hear such precise nomenclature as "mid-sized supermini" and "mini supercomputer" indicates just how hard it is to draw lines between the computing resources we have available today.

There is, in fact, far more of a continuum of price performance available in computers today than at any time in the past, and this is largely unappreciated by the buyers and sellers of this equipment alike. At the same time, the emergence of a small number of standard operating systems, languages, user interfaces, and networks has made migration between these machines unprecedentedly easy.

Here are three machines running AutoCAD. The first is the Data /indexData General One General One, a 9 pound laptop portable. The second is a PC/AT. The third is a Sun workstation running Unix. The program functions identically on all of these machines. Data may be moved between them using any of several media.

Only ten years ago, it was taken as commonplace that every computer vendor had its own operating system, data formats, and user interface. Users have demanded that industry standards be adopted to protect their investment in data, programming, and user training. They communicated this demand to the hardware suppliers in the most direct method possible, by choosing to buy only systems which gave them this protection. And the survivors among the suppliers have learned.

Let's look at how distribution of these products is developing. Back in 1983, if saying the words, "PC Software" failed to empty a room of venture capitalists, adding "Retail distribution" usually did the trick.

But the unavoidable fact is that as the price of computer systems and software falls, and the number and range of customers consequently expands, the way these products reach their users must change. You simply cannot

afford to sell a $2500 product to hundreds of thousands of customers with the type of direct sales force and on site support that has been characteristic of the computer industry in the past.

When we started Autodesk, we decided that we would not sell any products directly, but rather pass them through reseller channels. We initially focused on computer manufacturers, who were at the time building software distribution arms and targeting the major companies. Concurrently, we began to build a dealer network to sell our product to users and provide local support. We have seen an ongoing shift in the distribution of our product from a mix dominated by OEM sales to computer manufacturers in 1983 to one today where more than 85% of our product is sold through dealers.

Some people may not take computer dealers very seriously, but in less than four years, they've sold twice as many CAD systems as any of the CAD companies have in two decades.

Who are these dealers? Originally our dealer requirements were fairly low. If you bought two packages and paid COD, you were a dealer. We didn't go through the extensive qualification filtering that some of our competitors did, looking at square feet, location, staffing, service facilities, and so forth. Thank goodness. Some of our most successful resellers were those who wouldn't have been taken seriously by most software vendors. One, for example, was an architect who bought the package for himself and than began to configure systems for other architects. In less than three years, this architect has built a CAD sales, service, and support facility employing 14 people and selling to a wide variety of users.

Today, our dealers are much better qualified and trained. Every one of our more than 1300 dealers is required to have a full time person on staff with a drafting or engineering background. They are required to attend dealer training at our headquarters, and must return yearly for update training in new products and updates to the current product. Dealers must commit to providing direct user support for every package they sell.

Dealers who are successful in selling CAD and engineering applications are hard to characterise. Some specialise in the sale of CAD, while others carry a broad line of products. Our successful resellers include chain computer stores who have taken on CAD as an additional product line. Some focus on sales to one vertical market, such as architects or mechanical engineers, while others sell a basic drawing facility to a wide variety of users.

Further, while the computer retailer is often seen as a storefront selling to individual walk-in customers, many of our resellers have closed major sales with Fortune 1000 companies, government, and education by outreach marketing.

Many software vendors today are bemoaning the poor health of the dealer channel. We hear things said like, "how viable is a business where all of your resellers are losing money". Yet many of these companies who express concern for their resellers place their products in mass distribution channels which result in the package being sold mail order at prices at or below the dealer's cost. Is it any wonder that computer dealers don't spend much time mastering, demonstrating, and selling software?

If a software vendor wants to build and support a healthy dealer network, it must resist the temptations which destroy the vitality and profitability of that channel. It must decline to sell through discount channels. It must refuse to skim the cream off the dealer's business by selling direct to larger customers at a discount. And it must provide the training and support to keep the dealer competent to support his customers.

Many of those who said that CAD couldn't be sold through dealers pointed to the extensive user training and support traditionally provided with larger CAD systems. As desktop computers enter the engineering market, other collateral forms of support and training are evolving.

But nobody has as much control over how users come to master a product as the people who designed it. User interface is a distribution issue. If a product is sold for hundreds of thousands of dollars and used by a full time operator, it is reasonable for the vendor to provide a two week training course for the operators, and direct customer assistance.

When we designed AutoCAD, we knew that it would be bought and used primarily by users who had never seen a CAD system before. These first time users required a user interface that led them through the package, and extensive documentation, on line help, and support materials to aid them in getting started. In some ways, a CAD system for a first time, or part time user needs to be more complicated than one used by a full time CAD operator. The CAD operator has learned and invented work-arounds to achieve things the system cannot do directly. In addition, CAD operators usually have a manual drafting background and can always fall back on paper and pencil techniques where the CAD system lacks power.

For example, suppose the user has drawn two circles and wishes to draw a line perpendicular to one and tangent to the second. If the CAD system cannot perform this geometric calculation itself, the casual user is far less likely to know how to do it with construction lines as you did on the drawing board.

There is an aphorism in the electronics business that goes, “it costs a lot of money to make something cheap”. Well, in the software business it takes a lot of work to make something simple. In order to diffuse these previously specialised tools to communities of users thousands of times as large as before, we have to make major investments in user interface, on-line assistance, and teaching tools.

As a previously specialised tool becomes a widely distributed de facto standard, independent support becomes available to the users. The wise vendor encourages and promotes this. Autodesk has, for example, designated 43 organisations Authorised Training Centres. These centres, many of whom are regular educational institutions, provide direct user training courses, with several offering introductory through advanced courses focusing on specific application areas. By the end of this year, over ten books will be on the market based on AutoCAD. Only a product which generates sales in the tens to hundreds of thousands can economically support these developments.

What we are seeing in distribution then, is a shift from expensive, hard to use products sold directly in small numbers, to cheap, easy to use products sold in vast quantities. The user has access to the same, or better, training and support as before, but provided by a large number of suppliers in an open market, rather than directly from one supplier.

Now let’s look a little bit into the future. As guidelines as to what is likely to happen, it’s worth remembering a few pieces of what was conventional wisdom in 1983. It’s just as true today.

Desktop computers are being sold by the millions. This is a situation unprecedented in the human experience, equivalent to the development of a mass market for music created by the phonograph or for theatre created by motion pictures and television. Those who take advantage of this opportunity will do very well.

Most tasks are done today manually or by something very close. Anybody who thinks that all the productivity tools have been invented hasn’t spent much time looking at their workday.

We have made enormous strides in the last five years in making enormously complicated tools usable with no training. This is only way to sell to a mass market.

Far less than one movie in ten is a hit. Far less than one book in ten is a best seller. Why expect any better for software? But there *are* hit movies and books, and the winners in the software wars will be as enduring.

Picking the end of a trend in progress is one of the most difficult and least profitable activities one can choose to undertake. For forty years, everybody who bet on the price of computing power ceasing to decline has been dead wrong. So while I can't predict whether the PC will become the universal engineering workstation, I firmly believe that whatever ends up occupying that niche will cost less that today's PC. It is instructive, for example, to compare the specifications of the Atari 1040ST, which sells for less than $1000, to those of the proposed ideal engineering workstation of just two years ago.

The customers are a lot smarter than many vendors think. Suppliers who put the power to extend and adapt their products into the hands of their users and resellers unleash the creativity of these people to create products that directly benefit the supplier... and at no cost. Vendors who begrudge others who build successful businesses based on their products stultify those in the best position to make those selfsame products successful.

Find a case in the entire history of computing, where a closed architecture, proprietary system has defeated an open architecture system in the market.

Find a case where you cut the price of something by ten times and don't create a market far more than ten times larger.

Some people are still trying to take a software product and bundle it with hardware and sell it as a turnkey system for one particular job. If you undertake this, remember that you are trying to resell somebody else's hardware in direct competition to the computer manufacturer who made it. Good luck.

The long delayed but finally occurring advent of networks is going to make some major changes in the way people work, and create some major opportunities for software suppliers. Everybody's heard the phrase, "networks will enormously reduce the need for a large, central mainframe", and I believe it's true, but managing the kind of distributed database this leads to, and integrating this into an organisation will require solutions to many difficult technical problems that have been inadequately addressed to date.

Engineers do not work on drawings or documents. They work on projects. They communicate with others, and interchange design data at several different levels of abstraction. The computer has much to contribute to the entire process of engineering, and the gains as individual productivity tools begin to work together will be as great as those of the tools themselves.

• • •

The PC sector of the business is not a sideshow. It's the main event. This is where the battle for the entry level user is being waged. The software that establishes itself as a standard in the PC market will be carried upward to more powerful hardware as users migrate to their own best point on the price performance curve.

Today, the overwhelming majority of first time CAD users are using PC CAD. The overwhelming percentage of students who are learning CAD are learning on PC CAD.

Users will come to see the workstation hardware itself as little more than an appliance which delivers the productivity gains embodied in software. It will eventually be seen as making no more sense to ask, is a PC right for engineering work than to ask what is the best television for watching cop shows.

And in twenty years, somebody will show a slide of an IBM AT and ask, "remember when computers looked like computers?".

Thank you.

Super Programmers

In late 1985 it became evident that the growth of AutoCAD, the need to support new hardware, and the pressing need to develop new products was outstripping the capacity of our programming staff, still largely made up of founders of the company. In addition, we saw that in our rush to expand we had lowered our standards, recruiting people who did not always share our key developers' commitment to excellence. Having built what we thought was an ideal environment for programmers, it seemed to me that we should try to attract more high-productivity, broadly-talented people.

I wrote some ad copy which I hoped would select for people with the properties we were looking for. Thanks to the efforts of Mauri Laitinen, we finally ran the much-revised advertisement below starting on March 17, 1986. I date the return of the company to the highest standards of technical excellence and innovation to that date, his effort, and this advert.

Are you one of those rare software people whose productivity is hundreds of times above average?

Autodesk, Inc., the leader in computer-aided design, founded by people like yourself, invites you to join us.

We're The Best: You're The Best

Our company was built by people who never said, "I can't do that." If you're the person we're looking for, you'll be able to design, implement, test, and debug complex software, both alone and collaboratively. The code you write will meet the highest standards of efficiency, maintainability, and modularity. You'll know how to integrate changes in large, complicated programs, and you'll combine design and implementation skills with an intuitive feel for the evolution of the product as a whole and for its position in the marketplace.

You'll be able to find or develop the theory you need to get your job done. You'll be literate, and able to communicate complicated technical concepts in simple and readable language. Your work documentation will meet the standards of the best tech writers and be suitable for immediate inclusion in our user manuals. You'll be able to express yourself clearly and persuasively, whether in a design session or while speaking with prospective customers at a trade show. And you'll take personal responsibility for all your work, as a matter of course.

You'll care enough for the commercial success of your programs that you'll work effectively with marketing and sales people, contributing ideas to best promote the benefits of the products you'll be developing. You'll take an active interest in the work of other people in the company, and be willing to apply your expertise to help with their problems and develop their skills.

We Don't Want Less Than The Best

What will we do? We'll pay you more than anybody else in the industry. Your pay here can start as high as $60,000 and rise as high as your contributions justify. There's no ceiling on the pay scale for technical people here; you can earn $100,000 if you're worth it and prove it to us. We give our workers stock options that mean something. Unlike companies that look at options as a way of enslaving employees, we intend our options to let you share in the success you'll be helping to create. If we do our job, you won't want to leave. And since we're a public company, your options represent real stock with real value, not funny money.

We value productivity and excellence. We continually strive to minimize politics and bull. We couldn't care less about hours and personal style (except when you will know that it matters). Many of our key producers work at home. We don't care about your degrees and titles; we care what you've done, and what you can do in the future. We value people like you; after all, that's all this business really is.

What You've Heard About Us Is True

We're in the computer-aided design business. We sell only software. Our company was founded in 1982 by 14 programmers and we built it (with no venture capital and no debt) to a size of 215 people with offices in 5 countries and monthly sales exceeding $2 million. We sell a program called AutoCAD that runs on most MS-DOS 16-bit machines and some 32-bit workstations such as the IBM RT, Sun, and Apollo. In fewer than three years, we've sold more CAD systems than any other CAD company, micro, mini-, or mainframe. We've passed IBM and Computervision and we're pulling further ahead. We write exclusively in C (AutoCAD has more than 100,000 lines of it), and we develop a lot of our own software tools.

Our newer products include CAD/camera, a revolutionary raster-to-vector conversion program; AutoCAD AEC, a series of applications to dramatically speed up architectural and mechanical drawing and design; and AutoLISP, a full LISP interpreter built into AutoCAD. We do most development on our target machines. Relevant experience is nice, but if you're as good a person as we want, you'll be able to pick it up in a week or two.

We try to make this company the kind of place people you would want to work for. If you're the person we've described, we invite you to write us a letter describing what you can do, and what you've done.

During your interview, expect to discuss almost anything with five or ten other super programmers (three of them are on our board of directors). If you're a head-hunter, forget it.

Our company is continuing to turn the industry upside down. You can make a difference here and reap the rewards of your efforts. We couldn't find a company like this one four years ago, so we built one to our specs. Maybe they'll be your specs too.

CAD: the Heart of Computer Science

With the introduction of AutoLisp and the growing power of AutoCAD as a modeling system, it became clear to me that we were on to something far more powerful and significant than a drafting system. This was an attempt to place CAD in the position I believe it deserves—in the mainstream of computer science—as opposed to the backwater in which many believe it languishes. This was an internally-distributed "think piece". I am even more convinced now of the arguments expressed herein than the day I wrote them.

Computer Aided Design:
Vertical Market Application, General Purpose Productivity Tool, or the Heart of Computer Science?

by John Walker
Revision 1 — May 19, 1986

Over the brief history of Autodesk, we have observed the evolution of how CAD is perceived. We have always believed that we were selling a "word processor for drawings", suitable for anybody who draws as part of their work. The market as a whole and the analysts in particular, saw CAD as "a package for architects", or at most a tool applicable to a small set of highly specialised markets.

Time has proven us right. We could not have sold so many AutoCADs so rapidly, nor would we have the broad and flat distribution among market segments were CAD as specialised as the pundits believed. Moreover, the fact that have continually opened new, "nontraditional" markets for CAD without even trying vindicates our belief and confirms that in this case the users are way ahead of most of the sellers.

What we once knew and took action on is now becoming the conventional wisdom. Recently, in a more general context, Carl Machover wrote that the "computer graphics" industry is disappearing as it is assimilated into the mainstream of the computer industry. What we are seeing in CAD is part of this overall trend. Thus, the head start in positioning our product that this insight gave us is no longer a competitive advantage. We should look forward now to where CAD will evolve next (keeping in mind that it might not evolve anywhere, and that our original perception was the end of the road).

In this paper, I will suggest that what we have seen so far is simply the first step of an even greater integration of computer aided design into the mainstream of computer science and the computer industry. I will point to trends and events which, I feel, confirm this, and I will suggest product and marketing directions which will position Autodesk to take advantage of this trend.

What is Computer Aided Design?

I will bypass writing 50 pages on the various definitions of CAD, the acronym. Here let's consider what we mean when we say "computer aided design" itself. I propose the following definition:

> Computer aided design is the modeling of physical systems on computers, allowing both interactive and automatic analysis of design variants, and the expression of designs in a form suitable for manufacturing.

I think that this definition encompasses all of the types of work that is subsumed under the CAD umbrella, in all the various areas of application.

This definition implies that simulation is a far more important part of CAD than design description. I believe that this is true. Also, computer graphics has nothing at all to do with CAD, except as the servant of design, simulation, or presentation.

What Should a CAD System Be?

To best fulfill the definition of CAD given above, a CAD system should be a computer system that allows modeling of physical systems. To date, modeling has been done almost entirely with hard-coded dedicated systems usable only for one form of design: there's not a lot in common between ANSYS and SPICE. But, after all, the physical universe is a unified place with common rules, and it's not at all clear that one should have to write tens of thousands of lines of FORTRAN just to get started on a general-purpose modeler.

CAD systems to date have developed into general-purpose tools that understand geometry. From MacDraw to Medusa there is a continuum of knowledge about geometry and operations on either 2D or 3D primitives. What knowledge of reality exists is usually welded on as an afterthought (the very word "attribute" indicates how reality takes a second seat to geometric description).

A typical CAD system offering has a geometry processor with attached database, providing a "common design database". Analysis and simulation sits on top of this core, embodied in a host of separate programs which intercommunicate, if at all, only by passing information through the database. If one wants to create a new analysis program, "well, we have a FORTRAN compiler and library that lets you read the database".

Need it be this way? Can we not imagine a geometry-based CAD system evolving into a system which describes physical objects, and knows about the various ways in which they interact (and can be taught about interactions as we define new forms of geometry today)? Such a system would encompass all of what a CAD system does today, and would provide a common user interface and model for working with reality represented in a database.

What Does Simulation Have To Do With It?

Alan Kay, delivering the keynote speech at the Second West Coast Computer Faire in 1978 said, "we decided to focus on simulation in Smalltalk, because that's the only really interesting thing to do with a computer".

When I heard this, I was aghast: "Simulation", I thought, "why in the world would people want to use personal computers to model throughput in a machine shop, or to calculate the number of toilets[211] in a football stadium". Certainly any rational person wanted a personal computer to do real computer science on it: to write operating systems and compilers so that others could use them to write programs, and... well, I hadn't thought that out completely.

"Simulation" had come to mean (at least in the computer science lexicon), a specific kind of modeling of systems, usually done in an odd simulation language such as Simscript or Simula. What I only realised years later was that what Alan Kay was talking about something far more grandiose when he said "simulation": getting the whole wide world into that itty-bitty can: the computer. And yet, "simulation" in the limited computer science sense has already had a great and often little-appreciated impact on computer science as a whole. In his speech, Alan Kay exhorted people to look closely at Simula-67 for the direction of the future. Simula-67 included (in 1967!) classes, object orientation, multiple communicating processes, in fact close to a laundry list of what is currently considered the way to approach complex problems. So simulation in the small has already influenced the mainstream, and I believe that simulation writ large will have an impact many times greater.

What Do You Do To Get There?

It is time to start considering that we are in the business of bottling reality. If we accept the premises and conclusions of this paper, we should begin to undertake the representation of physical reality within our system, and provide the hooks which allow modeling and simulation to be added on to the system as easily as one adds geometric operations today.

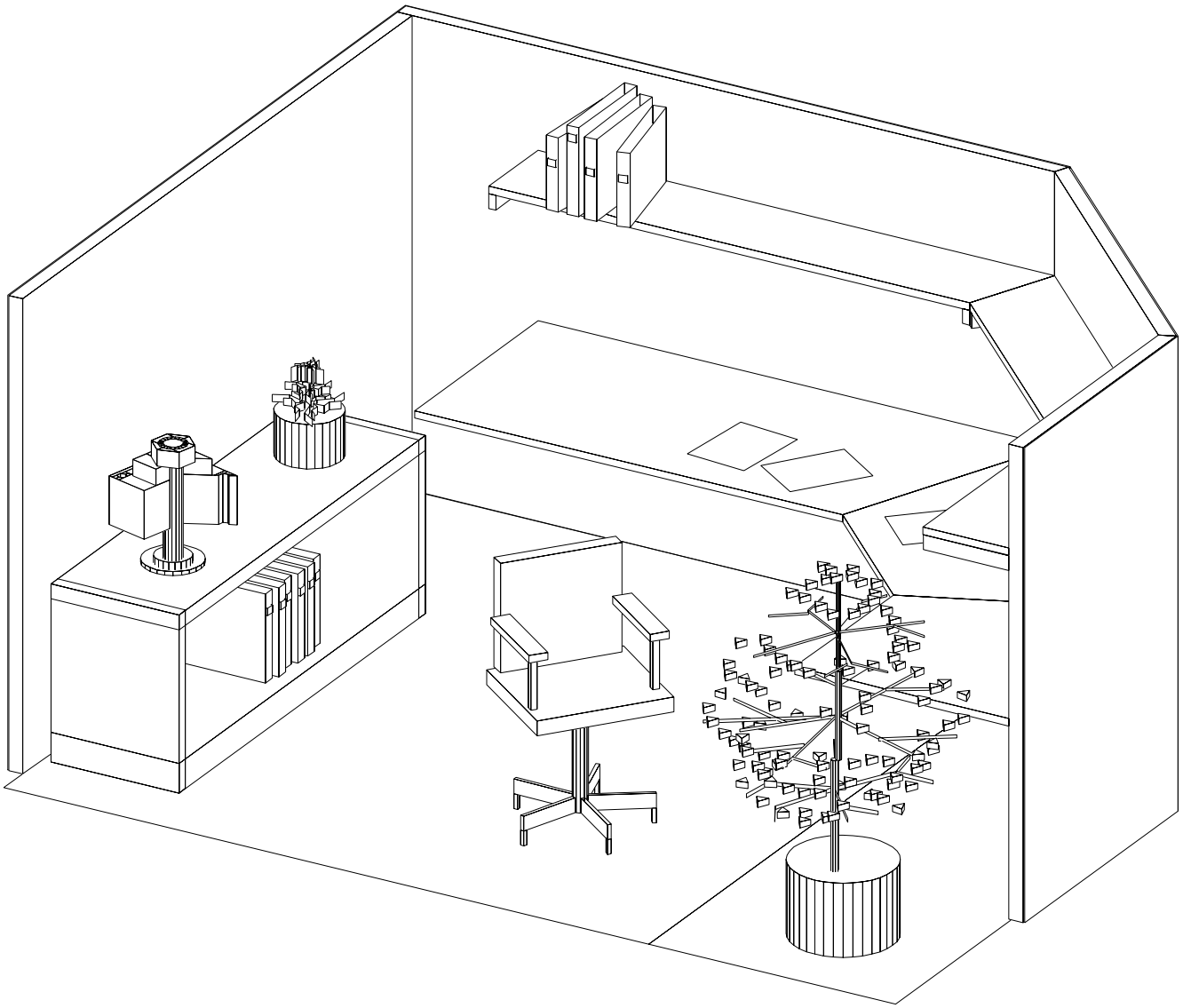
One specific example of what I'm talking about is the concrete proposal to add dimensionality to variables in our package. But this is just a first step on a road that may be decades long. I don't have the solutions and I don't even know many of the problems, but I'm becoming convinced that this is what this business is really about.

The Heart of Computer Science

As computers become more powerful, and tools evolve that allow us to build larger and more complex software systems on them, the artificial barriers that keep us from modeling the real world will fall. Finding the ways to do this best, and to build the systems that will be used for this is, I feel, what computer science is really about. And the company that knows this first and does the right things about it has prospects that are very bright indeed.

[211] But then, if there's a toilet, Autodesk *must* be involved.

This drawing, created by Gary Wells, was the original AutoShade demo drawing. Literally hundreds of bugs were found by this drawing during the development of AutoShade.



The Computer Revolution

I believe that the perceptions expressed herein point the way to the next generation of hugely successful computer products. The fact that I have been totally unsuccessful in explaining these concepts in numerous forums only reinforces my belief that they will eventually be considered self-evident.

The Computer Revolution, "What You See Is What You Get", and the Service Economy

by John Walker
Revision 6 — May 20, 1986

Abstract

The failure of personal computers to penetrate the office, engineering, and home markets as rapidly as many observers expected is a mystery to many in the industry. This failure is argued to be the result of a dissonance between the evolution of computer software toward placing decisions directly on the user, and the general trend in society toward services, intermediation, and division of labour. Suggestions of types of software products which are in harmony with the overall trends are made.

Background

The Computer Revolution

"You will have robot 'slaves' by 1965."
Mechanix Illustrated, cover, 1955

Whatever happened to the computer revolution? If you asked almost anybody in the computer business in the 1960's and early 1970's to project the consequences of the price of computing power falling four orders of magnitude, they would almost universally have seen the introduction of computers into almost every aspect of

life. Clearly, every home would have one or more computers, and computers would be used to communicate, write, read, draw, shop, bank, and perform numerous other tasks.

This was the "computer revolution": sales of computers in the hundreds of millions, with their diffusion throughout society making an impact on the same scale as that of the automobile, television, or the telephone. This forecast was so universally shared that all of the major players in the game bet on it and polished their strategies to end up with a chunk of the market. But in this revolution, something funny happened on the way to the barricades.

After an initial burst, sales of computers into the home market have slowed dramatically, to the extent that the companies that concentrate on home computers and software are considered pariahs. Penetration of computers into the office has been much slower than many projections, and the promises of networks and the "universal workstation" remain largely unfulfilled. The current dismal climate in the semiconductor and computer industries is a consequence of this. This despite the fact that the computer revolution has technically succeeded; now for a price affordable by virtually every family in the country, and by every business for every employee, one can buy a computer that can be used for communication, reading, writing, drawing, shopping, banking, and more. But they aren't selling.

What You See Is What You Get

"What You See Is What You Get" (abbreviated WYSIWYG, pronounced "wizzy-wig") has become the metaphor for most computer interaction today. It first became popular with screen-oriented text editing programs, became the accepted standard in word processing, and has been extended to graphics as exemplified by paint programs such as MacPaint and CAD programs such as AutoCAD. Today, this interface is being applied to integrating publication-quality text and graphics in products such as Interleaf and PageMaker, and is the universal approach to desktop publishing. Attempts to extend the WYSIWYG concept to encompass the user's entire interaction with a computer date from the Smalltalk system, and today are exemplified by icon based interfaces such as the Macintosh operating system and GEM. The generalisation of WYSIWYG to more abstract applications is sometimes referred to as a "Direct Manipulation Interface" (DMI).

The Service Economy

The evolution of an economy from one dominated by agriculture and extractive industries, through industrialisation, to one dominated by service industries is one of the most remarked upon events of our age. Our economy is moving from one that prospers by doing things to stuff to one in which we do stuff to each other. Not only does everybody believe this is happening, it really is happening. For example, walk through any major city in the US and observe that there are four banks on every street corner. If this were not a service economy, there would be, say, four machine shops.

As numerous pundits have pointed out, this evolution is natural and is not a cause for concern. We will all become so rich selling innovative financial services, fast food, and overnight delivery services to each other that we will be able to buy all the computers from Japan, televisions from Korea, and steel from Bulgaria that we need. That makes sense, right?

The Evolution Of Computer Interfaces

Let's look back into the distant past and see how peoples' concepts of how computers would be used have changed. Originally (I'm talking 1950's) it was assumed that a systems analyst would specify how the computer would be applied to a problem. This specification would be given to a programmer, who would lay out the control and data flow to implement the system. A coder would then translate the flowchart into computer code and debug the system, calling on the programmer and analyst as required. The coder's program, written on paper, would be encoded for the machine by a keypunch operator. After the program was placed into operation, it would be run by the computer operator, with data prepared by an operations department.

Well before 1960, the development of assemblers and compilers (and the realisation that the division was a bad idea in the first place), collapsed the programming and coding jobs into one. The distinction between programmer and systems analyst had become mostly one of job title, pay, and prestige rather than substance in many organisations by 1970. The adoption of timesharing in the 1960's accelerated the combination of many of the remaining jobs. Now, as any computer user could type on his own keyboard, the rationale for data entry departments began to disappear (while concurrently, development of optical character recognition and bar code equipment reduced the need for manual entry of bulk data). In the 1970's, one person frequently designed a program, typed it in, debugged it, ran it, entered data into it, and used the results.

The concurrent fall in the price of computer hardware contributed to this trend. The division of labour which developed when computers were used in large organisations was impossible and silly when a computer was purchased by a 5 man engineering department. This led to the "every man a computer user, every man a programmer" concept that lies at the root of the "computer literacy" movement.

The skills required to use a computer were still very high, and the areas in which computers were applied were very specialised and generally concentrated in technical areas where users were willing to master new skills to improve their productivity.

Word processing (and later the spreadsheet) changed all that. With the widespread adoption of the CRT, it was possible to build a user interface that immediately reflected the user's interaction with the computer. With this interface it became possible to build programs that could be mastered in far less time by users with no direct computer-related skills. The personal computer accelerated this process. Now a financial analyst could directly build and operate a spreadsheet model, ask "what if" questions, and obtain rapid responses. Now an individual could enter, correct, and format documents which would be printed with quality equal to that of the best office typewriters.

Attempts To Extend the Interface

The success of these two applications, which have accounted for the purchase of a large percentage of the desktop computers sold to date, led to attempts to make other applications as transparent to the user and thus as accessible to mass markets. Applications which paralleled word processing (such as CAD or raster-based drawing programs) were relatively easy to build and have enjoyed some success. Lotus attempted in 1-2-3 and Symphony to embed other functions within the metaphor of a spreadsheet, but one might read the market as saying that 1-2-3 was about as far as the sheet would spread.

The Macintosh user interface represents the most significant effort to date to create a set of diverse applications

that share a common set of operating conventions. Certainly the effort has gone much farther toward that goal than any which preceded it, but the limited success of the product in the marketplace may indicate that either the common user interface does not extend deep enough into the products to be of real benefit to the user, or that the user interface is still too complex. (On the other hand, it may indicate that people don't want a 23cm screen, or that Apple Computer should be named "IBM").

Today, desktop publishing systems are placing far more control and power in the user's hands than before. Now the user can enter text, then control its typography, layout, insertion of illustrations, and every other parameter controlling the final output. We see document filing and retrieval systems which have interfaces which emulate a physical library. Engineering programs are being built which graphically simulate physical systems and allow dynamic interaction with computer-simulated objects. We begin to hear the phrase, "this job is 5% solving the problem and 95% user interface".

It is currently almost an axiom of the industry that "people won't buy computers in large quantities until we make them easier to use". This is often used to justify the conclusion, "we must invest far more work in user interfaces, and make our programs more interactive and responsive to the user". I would like to explore whether this really follows.

"Go Do" Considered Shameful

I would like to suggest that the present state of computer user interfaces is the outgrowth of the postwar "do it yourself" culture, which peaked around 1960, and is not consonant with the service economy which began to expand rapidly around that date.

Consider how a business letter used to be written. The author would scribble some notes on a piece of paper, or blither something into a dictating machine. A secretary would translate the intent into English and type a draft of it. The author would then review it, mark up desired changes (which might be as general as "can you soften up the second paragraph so it doesn't seem so much like an ultimatum?"), and receive a final draft, which was signed and returned to be mailed. No wonder, as IBM pointed out in the early 60's, such a letter cost about $4 to prepare.

Now let's peek into the "office of the future". The executive's 50cm screen is festooned with icons. Picking the one with the quill pen opens up a word processor. The document is entered, pulling down menus to select appropriate fonts for the heading, address, and body. After the text is entered, a few more menu picks right justify the text. Dragging an icon of a schoolmarm into the window performs spelling checking and correction, allowing the user to confirm unusual words. The user then drags a rubber stamp icon into the window and places a digitised signature at the bottom of the letter (remember, you saw it here first). Finally, popping up the "send" menu allows the executive to instruct the system to send the letter to the address in the heading (electronically if possible, otherwise by printing a copy on the laser printer in the mailroom, which will stuff it into a window envelope and mail it), and to a file a copy under a subtopic specified by entering it in a dialogue box.

As we astute observers of the computer scene know, this is just around the corner and "can be implemented today on existing systems as soon as the users are ready to buy it". And what an astounding breakthrough it will be. An executive, who previously only needed to know about how to run a multi-billion dollar multinational corporation, analyse market trends, develop financial strategies, thread around governmental constraints, etc., now gets to be a typist, proofreader, typographer, mail clerk, and filing clerk. That makes sense, right?

Those involved in computing have discovered that doing it themselves is a far more productive way of getting things done than telling others to do it. They then generalise this to all people and all tasks. This may be a major error.

As we have developed user interfaces, we have placed the user in far more direct control of the details of his job than before. As this has happened, we have devolved upon the user all of the detailed tasks formerly done by others, but have provided few tools to automate these tasks. In the 1960's, engineering curricula largely dropped mechanical drafting requirements in favour of "graphical communication" courses. The rationale for this was that engineers did not make final drawings on the board, but rather simply needed to be able to read drawings and communicate a design to a drafter who would actually make the working drawings.

Now we're in the position of telling people, "you don't need all of those drafters. With CAD, your engineers can make perfectly accurate final drawings as they design". Yeah, but what do they know about drafting? And what help does the computer give them? The **`STRETCH`** command?

I have always taken personal pride in the statement "I don't ask anybody else to do my shit work for me". I am proud that I do my own typing, formatting, printing, and mailing. And as I have often remarked, I consider that I am better at doing these things than many of those paid full time to do them. I think that this is an outgrowth of the general "do it yourself" philosophy which goes all the way back to "self reliance". Although I often make the argument based on efficiency (I can do it faster by myself and get it right the first time, than I can tell somebody how to do it, redo it, re-redo it, and so on), the feeling goes much deeper than mere efficiency, and I think it is shared by many other computer types. I think that this is because those in the computer business are largely drawn from the do it yourself tinkerer culture. I do not have statistics, but I'll bet that the incidence of home workshops is many times greater among computer people than the general populace. Computers have, I think, largely absorbed the attentions of the do it yourself culture. Notice that the decline of the home-tinkering magazines (*Mechanix Illustrated*, *Popular Science*, *Popular Mechanics*) which had been fairly stable in content since the 1930s occurred coincident with the personal computer explosion and the appearance of the dozens of PC-related magazines.

As a result, we build tools which place more and more direct control in the hands of the user, and allow him to exert more and more power over the things he does. We feel that telling somebody else to do something for us is somehow wrong.

This shared set of assumptions is carried over into the way we design our computer-based tools. We build power tools, not robot carpenters. We feel it is better to instruct the computer in the minutiae of the task we're doing than build a tool that will go off with some level of autonomy and get a job done. So our belief that people should not be subservient to our desires may lead us to build software that isn't subservient. But that's what computers are *for*!

I'd ask you to consider: in a society where virtually nobody fixes their own television sets any more, where the percentage of those repairing their own cars is plummeting, where people go out and pay for pre-popped popcorn, and where the value added by services accounts for a larger and larger percentage of the economy each year, shouldn't computers be servants who are told "go do that" rather than tools we must master in order to do ourselves what we previously asked others to do for us? Recalling a word many have forgotten, aren't we really in the automation industry, not the tool trade?

Son Of Batch

Let's consider a very different kind of user interface. I will take the business letter as an example. Suppose you get a letter from an irate customer who didn't get his product on time. All you can do is apologise and explain how you'll try to do better in the future. You punch "answer that", and the letter goes away. The system reads it, selects a reply, and maybe asks you a couple of questions (queuing them, not interrupting your current task), and eventually puts a draft reply in your in basket. You mark it up, making some marginal notes, and maybe inserting a personalised paragraph in the middle and send it away again. It comes back, with your spelling and grammar corrected, beautifully formatted in conformance with the corporate style sheet. You say "send it", and it's all taken care of in accordance with the standard procedures.

Or, today, you pound out a 20 page paper with hand written tables referenced by the text and mail it to *IEEE Transactions on Computers*. Your typescript may be full of erasures and arrows moving words around. A few months later you get back the galleys, perfectly formatted to fit in the journal, with all of the tables set up and placed in the proper locations in the text.

Before you say, "that's very nice, but we don't know how to *do* that", pause to consider: *Isn't this what people really want*? If you had a system that did this, that could go off and do the shitwork, wouldn't you prefer it to all the menus and WYSIWYG gewgaws in the world? Also, remember that the computer would not be perfect at its job. Neither are people: that's why they send you the galleys. But isn't it better to mark up the galleys than to have to typeset the whole works yourself?

Building systems that can go off and do useful things for people is going to be very, very difficult. But so is designing the highly interactive user interfaces we're all working on. Both dwarf the effort involved in writing an old-style program where you just solved the problem and left the user to fend for himself. If you're going to succeed, you not only have to solve the problem well, you have to solve the right problem. Maybe improving direct user interaction is solving the wrong problem.

Building Subservient Systems

Can we build systems that do what people want done, rather than do it yourself tools?

There are some indications we're getting there. Autodesk has a product in its stable which may be the prototype of the 1990's application. I'm talking about CAD/camera. You take a picture and say "go make that into a CAD database". It goes and gives it its best shot, and you get to look at the result and change what you don't like about it, and teach it how to do its job better. Notice that of all of our products, this is the one people take least seriously because nobody is really confident that a computer can do what it sets out to do. Yet of all our products, it is the one most assured of success if it actually does a good enough job. This dichotomy characterises subservient products... look for it.

There is a database system on the market called Q&A from Symantec which lets you enter queries such as:

> How many people in sales make more than $50000 in salary and commissions?

then:

Which of them live in Massachusetts?

This really works. I have a demo copy. Try it.

I am not sure that anybody will ever actually build an "expert system", but much of the work being done in rule-based systems is directly applicable to building the types of products I'm talking about here.

Although I have only recently pulled all of these threads together, I've been flogging products like the ones I'm pushing now for a long time (so I have a bias in their direction). **NDOC** is the second word processor I've built which attempts to do reasonable things with very little user direction. The first one obliterated its competition, even though it provided the user much less direct control over the result. In the world at large, the battle between TEX and SCRIBE is being fought on the same ground.

When you encounter a subservient system, it tends to feel somewhat different from normal computer interaction. I can't exactly describe it, but try the "Clean up" operation on the Macintosh or the FORMAT command in Kern's editor and see if you don't understand what I'm saying.

Summary

The "computer revolution" has slowed unexpectedly. This is due to lack of user acceptance of the current products, not due to hardware factors or cost.

A large part of the current computer culture evolved from the do it yourself movement. This led to the development of computer products which require direct control and skill to operate. The computer industry is no longer involved in "automation".

The society as a whole has moved away from the self-reliant, do-it-yourself, ethic toward a service economy.

Until products which can perform useful tasks without constant user control are produced, the market for computers will be limited to the do it yourself sector: comparable to the market for circular saws.

Building such products is very difficult and one cannot be assured of success when undertaking such a project. However, it may be no more difficult to create such products than to define successful dynamic user interfaces.

Customers want to buy "robot slaves", not power tools.

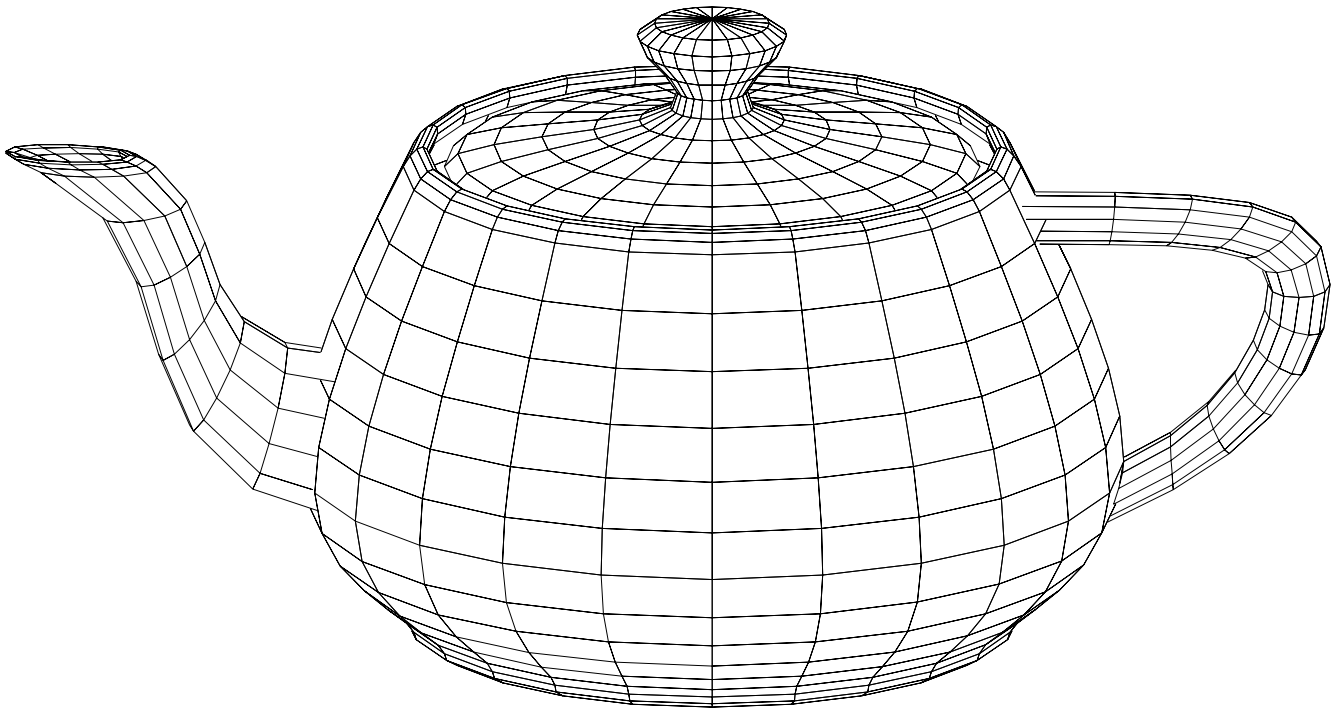
The market for such products is immense.

The Teapot is one of the classics of computer graphics. The teapot was originally hand-digitised as Bézier curves by Martin Newell, then a Ph.D. candidate at the University of Utah, in 1975. The control points for the Bézier surface patches were based on a sketch of an actual teapot on his desk. The original teapot now resides in the Computer Museum in Boston.

Newell's teapot became famous through his work and that of Jim Blinn. The teapot has become one of the standard test cases for any rendering algorithm. It includes compound curves, both positively and negatively curved surfaces, and intersections, so it traps many common programming errors.

For the full history of the Teapot, please refer to IEEE Computer Graphics and Applications*, Volume 7, Number 1, January 1987, Page 8, for Frank Crow's delightful article chronicling its history.*

I typed in the control point data from that article and wrote a C program to generate an AutoCAD script that draws the teapot. The teapot was one of the first realistic objects modeled with the three dimensional polygons introduced in AutoCAD 2.6, and has served as an AutoCAD and AutoShade test case ever since.



AutoBits

Humour has always been a popular outlet for the stresses and strains of Autodesk people. Here's a collection of some of the best.

To: Technical Staff
From: K. R. Throop
Subject: Job Titles

Date: 1 March 1986

I'm growing increasingly concerned about the title escalation going on around here. It seems like every time we hire somebody we have to create a new job title to wedge them in the hierarchy without wounding somebody's ego.

This is hardly in the spirit of solidarity we shared when we marched in the Programmer's Strike For Parity twenty years ago.

Therefore, I suggest that we establish the following job titles for the technical staff, banishing "Software Engineer" to the darkness whence it came.

Programmer
Enhanced Programmer
Super Programmer
Ultra Programmer
Virtual Programmer
Senior Programmer
Elder Programmer
Doddering Programmer
Intergalactic Exalted Cosmic Hyper Programmer

Titles shall be unrelated to pay, and shall be chosen by the employee.

Kelvin

Kelvin R. Throop
Virtual Programmer

Autodesk Founder Announces New Product, "Hurls Down Gauntlet To Other Software Vendors"

Sausalito, California, October 1, 1986.

John Walker, President and a founder of Autodesk, Inc., makers of the popular AutoCAD drafting software today announced that shipments of AutoSketch, the company's $79.95 drafting software were commencing.

"AutoSketch delivers, for only $79.95, far more capability for 2D drafting and design than the $1000 AutoCAD we were selling only four years ago. The software industry continues to lead all others in delivering value for the dollar, and we're proud to be both the value and performance leader in our market", Walker said in a press conference announcing the event.

Concurrent with the AutoSketch announcement, Walker revealed that he had undergone a battery of tests and been pronounced "not insane". Walker said that he was examined by a panel of prominent Marin County psychiatrists, psychologists, faith healers, astrologers, and vacuum cleaner repairmen who rated Walker's sanity on a scale of 1 to 5. Mass murderer Charles Manson was used as the standard for 1, and physicist Werner Heisenberg was used as the standard of somewhere between 4.9 and 5.

The panel rated Walker as 2.6 on this scale, which Walker claims definitively demonstrates his mental soundness. In explaining why he was disclosing this information, Walker said, "when you buy computer software, you have a right to know if it was written by a nutcase or not. Norbert Weenie, adjunct professor of cybernetic chiropractic at Ukiah Community College has demonstrated convincingly that software can drive you crazy. And software designed by a loonie can make you just as bad. Inspector Harry Callahan of the San Francisco Police Department confirms this, saying, 'ever since that Goddam WordStar, you walk into one of the typist dives on Montgomery Street and say Control K, and it's worth your life'."

Walker immediately challenged other CAD system designers to similarly prove their sanity. And if they don't, he added, "well, then it's up to you the customer to ask 'what do they have to hide?' ".

Autodesk, Inc. develops and markets computer software for technical professionals. Products include AutoCAD, AutoSketch, CAD/camera, and AutoCAD AEC. Autodesk stock is publicly traded on the NASDAQ national market system under the symbol ACAD. John Walker is privately traded between the Securities and Exchange Commission, the Internal Revenue Service, and the Franchise Tax Board under the symbol 217-50-0239.

This is not really an AutoBit. It was a serious proposal which never came to fruition at the time, possibly because it was confused with an AutoBit. In March of 1988, the Autodesk Technology Forum was inaugurated, which essentially implemented this idea. This was one of the first programs created by the newly-founded Autodesk Research Lab. Speakers at the Technology Forum included in-house developers demonstrating products, vendors showing their wares, and luminaries from within the industry and outside including Timothy Leary, Stuart Brand, Alvy Ray Smith, Hans Moravec, Rudy Rucker, Jaron Lanier, Todd Rundgren, ancient DNA expert Svante Paabo, astronaut Michael Lampton, and Indianapolis 500 driver Randy Lewis. The Autodesk Advanced Technologies Department, successor to the Autodesk Research Lab, was disbanded on August 18, 1992, bringing to an end the weekly Technology Forum.

Autodesk Technical Seminars

Proposal by John Walker
Revision 3 — February 12, 1986

We all wish that more people in the company saw the big picture, took an interest in the operation of all facets of the company, and were able to help out wherever help was needed. Let's do something active in this regard rather than just moaning. I propose we establish weekly technical seminars, commencing at 17:30 each Thursday, in which somebody from Autodesk will hold forth for an hour to 90 minutes on some topic known in depth to the speaker. A list of suggested topics follows (in no particular order). I'm sure you can think of speakers who'd love to blither on them.

Introductory LISP programming
Implications of antitrust law on Autodesk
Autodesk history
Stock options—what they mean, how they work
How to close a sale
Digital electronics in one hour
What it's like to be a computer dealer
How bugs are fixed (and how to report them)
Fundamentals of plane geometry
What does an architect do?
Lock picking
What is CAD/camera good for?
Project management with PERT
What is the board of directors?
BASIC programming
How to use Knowledge Man
Data networks
Principles of quality control
How Autodesk's products are manufactured
What is a spreadsheet and why you should use one
Copy editing
International shipping: strange customs in faraway places
RS-232: How to make it work
Getting the most from Compu-Serve
How national accounts select products

Technical analysis of stock charts
What is an expert system?
How accounting works
The Smalltalk paradigm—user interfaces
How to write a contract
Creating printed material—concept to press

I'm sure that this list will serve as a springboard for many additional ideas. If we decide to do this, I intend to circulate a list to everybody of these topics and allow people to volunteer for any topic they think of. Presumably we can fill up the calendar rapidly.

I wasn't sure I liked the tacky music we used at the opening of the audio tape we used to enclose with AutoCAD. The week we recorded it, I produced an alternative opening with this script and music you can probably imagine. We didn't use my version.

CAD—The Final Frontier

CAD, the final frontier.

These are the voyages of the starship Autodesk.

Its five year mission—to seek out pockets of profitability in the CAD industry—and empty them.

To develop strange new products, and patiently explain them to bewildered analysts.

To blindly go where no venture capitalist has gone before.

Marinchip Defeats IBM PC/AT In Benchmark

Mill Valley, California, Mayday 1986.

John Walker, President of Marinchip Systems Ltd., announced today that the Marinchip 9900-based PC/OT (Personal Computer/Obsolete Technology) resoundingly defeated the IBM PC/AT in an intense floating point benchmark, even though the PC/AT was equipped with the 80287 math coprocessor.

The benchmark was an optical ray tracing program involving primarily floating point computations, including evaluation of trigonometric functions. The Marinchip 9900 PC/OT executed the program in 69.32 seconds, while the IBM PC/AT took 93.79 seconds to execute the same program.

"Our PC/OT executed this real-world engineering program 26 percent faster than IBM's much vaunted PC/AT, even though our 9900 processor was operating at 2 megahertz, one third the speed of the PC/AT's 80286 CPU, and the fact that the PC/OT was emulating floating point in software instead of using a mathematics coprocessor. This benchmark vindicates our RISC (Rinkydink Instruction Set Computer) architecture, and clearly demonstrates the superiority of our proprietary QBASIC language for scientific applications.", said John Walker.

The IBM PC/AT benchmark was run in Lattice C version 2.14, using the "`-p`" memory model (large code, small data). The standard Lattice 2.14 library was used. The results calculated by the Marinchip PC/OT and the IBM PC/AT agreed to 15 decimal places.

Commenting on the results, California Governor George Dookmayjeun said, "It just goes to show you how a bunch of clean living Californians can beat the spit out of those drug-soaked greasy Florida scumbags. Look, I don't give a flying fork what you quote me as saying, but please spell my freaking name right!".[212]

[212] This paragraph is made up. All the rest is the absolute truth. Honest!

This is for real. It was actually sent.

Mr. James Meadlock
President
Intergraph Corporation
One Madison Industrial Park
Huntsville, AL 35807-4201

9th July 1986

Dear Mr. Meadlock:

I was recently asked by one of our customers to comment on the attached memo from Mr. Ken Bado of your company. Since the memo contains a serious misstatement of the facts regarding our product, I felt I should bring it to your attention immediately.

AutoCAD maintains all of its coordinates internally in 64 bit floating point, adhering to the IEEE standard for double precision floating point (IEEE Std 754-1985). This delivers more than fourteen decimal places of accuracy. No physical constant is known to an accuracy of fourteen decimal places. That accuracy is more than adequate to position objects anywhere in the entire solar system (encompassing the orbit of Pluto) to within one centimetre.

Mr. Bado does not grasp the distinction between decimal places and bits. As I understand it, Intergraph is an integer-based system with 32 bits of accuracy. This provides for an accuracy of 4.2 billion units and a dynamic range from zero to 4.2 billion. AutoCAD's IEEE standard floating point provides a 52 bit mantissa, allowing an accuracy of 4.5 quadrillion units. This is more than a million times as accurate as Intergraph's system. In addition floating point gives us a dynamic range of plus or minus 10^{308}, which is over one billion billion billion

Eric Lyons wrote a number of papers intended to educate Autodesk folk in the intricacies of the three-dimensional world we were entering, both through the ongoing process of adding three-dimensional features to AutoCAD, and by our evaluation of ways to enter the solid modeling arena which finally led to the Cadetron acquisition (see page 355). Eric ran some of these papers through **TRAVESTY**, *the random language rearranger, and ended up with this. Another travesty of 3D appears on page 329.*

Yet Wah[213] 3D

Eric Lyons
9/21/86

In this chapter we'll learn about the combination of it, but it is fundamental to the way a solids modeling, anyway? Well, it's mostly a matter of representation in them, you'll like a fancy alloy automobile wheel. You end up with a CSG systems. CSG solids modelers work a little difference between a solid (with some user interface modelers). In fact, many B-rep system, you'll be able to *simulate* a hole in it with these problem—designing something called *interference checking*, and half of it, but it is fundamental engineering problems (some say it really start getting into another primitive and mash it together it lies outside, on the end. You modelers don't solve anything), but it is fundamental types of solids modeling systems in use today, primarily because they interface or off (outside) the same space, if you take a cylinder from a surface modelers. In fact, many B-rep modeling system works by taking little shapes. Each of this chapter we'll learn *any* of this chapter we'll learn about there are the point, but it does have a few shortcuts. Also, surface, or *inside* an object, you are created a solid. With a surface model.

[213] Yet Wah was a not terribly good Chinese restaurant in Sausalito

To: Kathleen Doney
From: AUTODESK SPINE POLICE, Kelvin Throop, Sergeant

Docket number: MDASMAN001.00

Date: 24 November 1986

The AutoSketch User Guide does not have its title on the spine.

The Autodesk Spine Police were established in 1983 by a directive from the Board Of Directors and empowered to take all steps necessary and proper to insure that every book Autodesk publishes has its title on the spine.

To date our record is crummy.

You can help.

Could we put “AutoSketch User Guide” on the spine in the next edition?

Thanks.

First VHSIC-Based Commercial Product Stuns Industry

* * * For Immediate Release * * *

Mill Valley, California: September 22, 1986.

In a move that astounded industry analysts, a previously unknown company in Marin County, California announced the first commercial product embodying technology developed by the Pentagon's Very High Speed Integrated Circuit (VHSIC) project.

Kelvin R. Throop, spokesman for Strategic Weapons Systems of Marin, Inc. announced VHSIC-CALC at an impromptu press conference held at the company's headquarters.

Throop described the product as a MIMD parallel processor composed of over one million Gallium Arsenide VLSI single chip processors. The processors form a 1024×1024 cellular array: each processor is connected to its four neighbors in a rectangular grid. All processors run off a common 500 Mhz clock. The processors are based on RISC architecture, and execute instructions from a 1K by 32 bit on-chip RAM. Most instructions execute in one cycle, resulting in a throughput of 500 million instructions per second per processor, or 500 trillion instructions per second peak system performance.

Each CPU in the array executes a control program which repeatedly evaluates a mathematical formula stored in the RAM. Values required by the formula may be stored locally or derived from other processors, and values calculated may in turn be routed through the array to other CPUs. Throop said that cellular arrays of this form have proved useful in many forecasting, analysis, and planning functions. The unprecedented throughput of VHSIC-CALC will make applications previously undreamed of possible, he claimed.

Defense and other government sources were quick to endorse VHSIC-CALC. General William Tecumseh Chaos of the Larkspur Nuclear Weapons Dump said, "in VHSIC-CALC we see the fruits of the billions we've pissed away on advanced technologies. Before, we often relied on guesswork to make our crucial decisions, but you know: a megaton here, a megaton there, and before long you're talking a really big hole. VHSIC-CALC will let us plan and forecast and anticipate the unknown with a clarity we haven't seen since Pearl Harbor". Reaction in Washington was enthusiastic: Quentin Terabuck, director of the Strategic Deficit Initiative said, "American technology is the key to America's security. Let's see those sneaky wiretapping commie slimebags recalculate their budget in 50 nanoseconds. And they talk about central planning!".

Industry reaction was muted. John Walker, president of Autodesk, Inc., another Marin County high-technology company, said "I applaud SWSOM's commitment to the mass market as illustrated by their choice of the Commodore 64 as the control processor for VHSIC-CALC. While I feel that the suggested retail price of 40 billion dollars will slow acceptance in the retail market, inevitable price reductions may lead to a growing presence in the VAR channel". When asked, "What's a VAR?", Walker expanded, "Beats me".

Here's another attempt by Eric Lyons to clarify a 3D document by running it through **TRAVESTY** *(the first is on page 326). I'm including this one as well because I gained an enormous insight from reading it—for years I'd wondered why I had so much trouble understanding orientation in three space. Upon reading this, it finally became clear. I was just an interpretation! Thanks, Eric.*

What is 3D is What

Eric Lyons

Today.

What is, the more the systems (AutoCAD or any other productive by using drawings for describing to someone has to interpretation that a concept). Nor is it a concept. Nor is it a concept of us live, virtually all in itself. Throughout the expensive CAD systems are inherently 2D—you're simply an interpretation. This process is, naturally, we will have been around since the real world (there are many people that make $20,000–$25,000 (including overhead) per year). But many people don't want it that way. You see, therefore the extra 'D' is for their draftsperson becoming more products. It is intended as an education of AutoCAD was introduced for 2D applicated the people want it that is, they were purchased based on this process is, naturally, extremely error prone. And it was good. So people bought a lot of engineering applications do not require anything things like Auto-trol, CV, Intergraph, and among ourselves. Soon everything is clear; a lot to interpretation and draftsperson, his annual salary, and them based on a sheets of paper is not a listing of features for the IBM PC. It didn't do nearly 80's, they were used to do just about. Now, in the complications. And drawing an interpret the early what a lot could be considered exceptions, (several sheets) so that could do for 2D applicated the popularity of 2D CAD systems *document* a physical system in placed on a sheets of greater accuracy (more on this paper is not a listing of features for their drafting) have something. Buildings, cars, stereos, cuisinarts, socks, airplanes, are all manufactured exceptions, (several sheets) so that a lot of people that cost justification of AutoCAD or any others who offered only primarily 2D—you're simply drawings. Thus the point that is 3D, anyway? Eric Lyons 9/17/86. There lies the fireplace the complicated the more consistent drawings are all manufacturing the system, let's start by discussing what a concepts and explain who wrote and the systems.

This gem was forwarded, on behalf of the author, in late 1987 by Lew Goldklang, whose resemblance to Mr. Dithering is striking.

AutoShield

From its secret corporate bunker just north of Sausalito, Autodesk today announced the release of AutoShield™, an exciting new software product that renders nuclear weapons impotent and obsolete. AutoShield runs on IBM PC, XT, AT, or 100% compatible computers. At least 640KB of random access memory is required; a math co-processor is recommended to ensure appropriate response time to incoming warheads.

The company also announced the formation of a new subsidiary, Auto-da-fé Inc., based in Key Largo, Florida, which will handle all production and marketing. Officials vehemently denied cynical press allegations that this would also provide a convenient front for money laundering operations out of Miami and Havana.

The new software defends against land-based SS–20 Malenkov-class ICBM's, submarine-launched missiles, cruise missiles, and aimlessly-tossed empty Stolichnaya bottles. It also protects against all suitcase weapons larger than a thermos of borscht.

AutoShield offers three levels of Advanced Defense Extensions which provide escalating levels of protection:

ADE 1: Passive—incoming projectiles bounce harmlessly off an invisible screen.

ADE 2: Active—allows users to shoot down second-stage boosters at ranges up to 280 nautical miles. Enemy rockets appear interactively on-screen like an arcade game.[214]

ADE 3: Aggressive—protects completely against a hostile first strike, and then takes the initiative to launch an all-out counterattack by broadcasting an extremely-low frequency (ELF) signal to all submarines lurking in the Bering Strait and all B–52's on patrol between your workstation and Thule, Greenland.

Ray Tracing, sightless yet visionary jazz pianist and company spokesman, who once pioneered dramatic advances in Moiré pattern analysis by counting LISP parenthetical nesting for 96 straight hours (just before he went blind), noted that AutoShield preserves a great American tradition, allowing each rugged individual to have as much or as little security as he pleases, without depending on his neighbors or any incompetent, flabby authority for protection against treacherous communist savages.

Mr. Tracing declined to give specific details on the revolutionary new technology, calling it proprietary information, but hinted that in the passive mode AutoShield uses a subroutine-activated piezoelectrochemical quark excitation tunneling technique—a variant of the GARDOL shield popularized in the Colgate toothpaste commercials of the late 'fifties. Apparently, scrapings from Ed Herlihy's teeth were crucial to its success, although hexachlorophene does not seem to be an active ingredient. He also pointed out that in both of the active modes AutoShield generates well over 10^{12} joules of energy out of seemingly thin air by new breakthroughs in sub-molecular resonance and unified field reverberation. "We just exploited a loophole in the second law of thermodynamics big enough to drive an Abrams tank through, before it rattles itself apart", chuckled the irrepressible Ray.

He also said that Autodesk engineers had tapped a hybrid technology that is a combination of harnessing the universal "strong" force binding together atomic nuclei, a previously unknown new force, believed to be the

[214] I tried this myself. Talk about fun—it was love at first strike.

mysterious "force" in the Star Wars movies, and the primitive metaphysical power of snapping one's fingers continuously to keep wild hordes of elephants away. "See, it works!", beamed Ray, snapping his fingers smartly while chewing gum at the same time, in a clumsy but evocative Gerald Ford imitation. "No elephants here, and no Russki missiles, either!"

"It was pretty trivial", claimed Ray. "One of our guys did it over the weekend out of sheer boredom, in between watching the football game and counterfeiting food stamps."

While a closely guarded corporate secret, the new technology seems to be quite similar to that used in the movie E.T., whereby a coat hanger, umbrella, tin foil and scraps of other crude, household throwaway items are employed to project energy beams across the vastness of all known galaxies and out to the very edge of the universe with a focus and intensity millions of times that of the most powerful industrial lasers.

Early beta-site versions of AutoShield required an additional minimum peripheral hardware configuration to support Advanced Defense Extension 3, including a Raytheon Patriot-class phased-array radar, a Titan three-stage booster rocket, a Shiva-class nuclear-powered X-ray laser and 10.2 kilograms of Uranium 238, enriched to at least 10% Uranium 235. In a last-minute, dramatic marketing maneuver, Auto-da-fé removed this restriction, calling it an expensive hardware crock. "By performing all counterforce measures in software instead, we are making AutoShield affordable to a much larger market of keenly defense-minded but nearly broke citizens", beamed backup company spokesman Mo Zambique. Major hardware vendors at Livermore Labs and Helionetics were chagrined at the loss of potentially lucrative OEM contracts, but vowed to recoup any lost business in the robust third-world counterinsurgency arena instead.

AutoShield is guaranteed to work during any level of attack, from restless native uprisings to small-bore tactical field weapons, and on through strategic counterforce measures all the way up to massive, all-out screaming Armageddon.

In the extremely unlikely case that it should fail, the company offers full credit toward another product, less a nominal fee for handling and postage. Proof of purchase is required, along with a comprehensive, well-documented bug report isolating the problem down to the suspect procedure within 10 lines of code.

Auto-da-fé is negotiating with Federal officials to sell a site-license to Washington for an amount equal to the entire defense budget. Government sources indicate this will be approved readily, as all other items in the new defense budget will also be rendered impotent and obsolete. "What the hell, we were going to squander it all anyway", said Pentagon spokesman Sergeant Fleakiller, "We'll just gouge it out of the acquiescent tax base one way or the other; there's still plenty of plasma left in that soggy turnip. No problem."

The imminent sale should give a one-time surge to the company's cash flow, resulting in new quarterly earnings per share of $35,714.29 which, at the present P/E multiple of 41.3 would indicate a fair market price of just over $1,475,000 per share. On Wall Street, respected securities analyst, swashbuckling international commodity trader, and incessantly blabbering TV game show host Long June Silver was euphoric. "It's really the ultimate product; every government on the planet will have to buy one. The company is going to absorb every last piece of currency in circulation. We liked the stock at $50 and continue to be aggressive buyers at these levels", he said expansively. Back in Marin, company officials quietly contemplated a 25,000-for-1 stock split.

Auto-da-fé is insisting on taking payment in newly-minted $7\frac{1}{2}\%$ Treasury bonds with a unit face value of $1,000,000 (the ones with Teddy Roosevelt's loudmouth blustering face on them). Company analysts are eager to scrutinize the fine engraving details and rag content. The Bureau of Engraving and Printing will have to run their recently acquired high-speed German presses night and day non-stop for nearly two years to print it all up.

Company spokesmen indicated that profits from the sale would go toward restocking the kitchen refrigerator with Classic Coke, which has recently dropped to levels so dangerously low as to impact programming productivity.

About the Author

Preston McChesney Dithering is a member of the Senate Foreign Relations Committee, the Federal Open Market Committee, a part-time editor of *Trialogue*, the Trilateralist newsletter, and an ephemeral, shadowy character who contributes these sporadic screeds about national security and economic issues for your personal edification; you shouldn't have to act so stupid. He is also a lifetime ex-officio member of the California Prune Advisory Board.

Mr. Dithering has written extensively and with extraordinarily sensitive insight on American finance, culture, and nutrition, including the phenomenal best-selling trilogy, *A Nation of Quote Suckers*, *A Nation of Screen Suckers*, and *A Nation of Aspartame Suckers*, available immediately through Authorized book dealers, value-added resellers, wholesale discount distributors, gray marketeers, anonymous mail-order houses, sprawling concrete suburban malls, chi-chi boutiques, supermarket checkout stand blister packs, home shopping networks, all-night mini-marts, pornographic newsstands, remote trading posts, bartering clubs, penitentiary commissaries, broken, ornery vending machines, and roving flatbed truck hawkers in your area. Watch for these profound works soon to be nationally televised as a blockbuster miniseries for your entertainment, simultaneously coordinated with a wave of life-threatening interactive battle toys, moronic video games, software simulations, bubble gum trading card series, sugary breakfast cereals, carcinogenic soft drinks, teasing puzzles, smart-ass T-shirts and other overpriced designer clothing, and tax-privileged theme parks in targeted redevelopment enterprise zones in your neighborhood.

Meanwhile, Mr. Dithering would really like to hear from all you bored housewives out there.

AutoShield is a trademark of Auto-da-fé, Inc.

Auto-da-fé is a wholly-owned subsidiary of Autodesk, Inc.

Auto-da-fé is a registered trademark of the Spanish Inquisition. Great fun while it lasted; too bad you missed it; coming again real soon to a theater of operations near you.

Bicameral legislature, self-evident truths, neo-mausoleum architecture, flatulent filibustering, pork barreling, ineffective protectionist tariffs, tobacco subsidies, Prohibition, paternalistic socialism, benign neglect, incomprehensible policies, doctrinaire intransigence, language obfuscation, currency debauchery, unbacked fractional reserve notes, hyper-elastic debt ceilings, debt monetization, confiscatory taxes, perverse Byzantine lopsided tax codes,[215] massive private gain at public expense, laissez-faire mercantilism, lucrative patronage, cozy cronyism, bottomless corruption, strangulating bureaucracy, comical ineptness, flippant insouciance, sullen insubordination, terminal bloat, creeping disintermediation, staggering trade deficits, voodoo economics, dollar bashing, intrusive citizen surveillance, satellite reconnaissance, calculated disinformation, shameless media hype, patriotic symbol manipulation, slush fund diversion, Senate select sub-committee investigations, lawyer-mongering, plea bargaining, subpoena quashing, pathological perjury, moral turpitude, ineffectual Presidential task forces, token civilian control of the military, forced conscription, sub-atomic tinkering, pitiful giantism, titillating indiscretion, stupefying horror, intermittent assassinations, lugubrious motorcades, virulent anti-communism, unenforceable loyalty oaths, manifest destiny, eminent domain, gunboat diplomacy, expeditionary brigades, insurrection squashing, gratuitous

[215] "One is, therefore, in fairly sedate baby-kissing company if one says (perhaps overcautiously) that the tax structure is a pullulating excrescence negating common sense, a parody of the gruesomely ludicrous, a surrealist zigzag pagoda of pestilent greed, a perverse thing that makes the prerevolutionary French system seem entirely rational."—Ferdinand Lundberg, *The Rich and the Super-rich*, Bantam Books, Inc., 1968, Page 391.

violence, gun-running, strategic incursions, surgical strikes, rural pacification, economic destabilization, blundering interdictions, horribly botched bizarre clandestine operations, official denial, mock bewilderment, outrageous interminable scandals, internecine power struggles, malignant brain tumors, bungled suicide attempts, affable doltishness, ludicrous gaffes, howlers in elementary logic, preadolescent fantasies, weepy nostalgia, technology worship, doomsday scenarios, emergency broadcast systems, self-righteous posturing, religious zealotry, institutionalized gibberish, stratospheric hyperbole, vituperative castigation, fulminating rhetoric, incoherent ranting, thundering bombast, pseudo-Evangelical blather, hair-splitting equivocation, bald-faced lying, wretched excess, moon golf, industrial disease, and bankruptcy are trademarks of the United States Government, 1913–1987. All rights reserved, or you'll hear from this week's Attorney General, who'll send the Feds down to slap the cuffs on you right in your office, deliberately embarrassing you in front of your own staff, just like on Wall Street. No fooling.

The Guts of a New Machine

From: Kathleen Marvin, Mon Feb 15 09:50:02 1988
To: tech
Subject: Inventing a machine

I need some help with some 6th-grade homework. I've been doing OK all year (except for a few problems with advance math), but finally have something that's got me and my daughter really stumped. She has to make an invention that is a "complex machine" made up of the following simple machines:

1. pulley
2. wheel and axle
3. lever
4. wedge
5. inclined plane
6. screw

She has to have the concept by tomorrow for her science class, and then has to draw it three-dimensionally. (No, there's no time to teach her AutoCAD!) Later, she actually has to *make* it.

Any ideas??

From: John Walker, Mon Feb 15 10:47:04 1988
To: kmarvin, tech
Subject: Re: The heart of a new machine

≫ 1. pulley
≫ 2. wheel and axle
≫ 3. lever
≫ 4. wedge
≫ 5. inclined plane
≫ 6. screw

All of them?? I know of few real machines that use all of these fundamental machines.

But what the heck... mechanical design was never my strong point, but who can resist a challenge?

The following is a machine to preserve domestic tranquility by reminding a programmer when s/he/it[216] has been at it too long and it's time to wander upstairs and see if one's spouse is still around.

When one starts to program, light a candle. The base of the candle has a string molded into the wax so that when the candle burns down to that point (after about 20 hours, a reasonable programming interval), the string is released.

[216] See page 256.

The string passes over a pulley (1), and the other end is connected to the long end of a lever (3), the short end of which is in front of a small cart placed at the top of an inclined plane (5). When the string is released by the candle, the weight of the long end of the lever causes it to fall, raising the other end of the lever and releasing the cart.

The cart rolls down the inclined plane on its wheel and axle (2). The bottom of the inclined plane is adjacent to the programmer's keyboard, and the small end of a wedge (4) is placed under the keyboard. As the cart rolls down the inclined plane, it acquires kinetic energy which is expended when it reaches the bottom and strikes the wedge. This drives the wedge under the keyboard, flipping it over, and reminding the programmer that programming is only part of life, albeit the best part.

The screw (6)? That's to drive into the ear of science teachers who make up dumb problems like this.

Portions of this design may be subject to patents issued or pending which are the property of the Turbo Digital Research Foundation, a division of the International Communist Conspiracy.

From: Dan Drake, Mon Feb 15 12:05:15 1988
To: kmarvin tech
Subject: The braining of a new machine

1. pulley
2. wheel and axle
3. lever
4. wedge
5. inclined plane
6. screw

The teacher pulls a rope, on the other end of which there's a weight, the rope going over a pulley (1) in between. The axle (2) that the pulley is on rotates (making the pulley also a wheel (2)). The end of the axle is threaded (6) and moves a nut as it rotates. Attached to the nut is a rod that runs parallel to the axle; on the end of the rod is a wedge (4) that's on its side, as it were, so that it pushes things to the right or left as it advances. What it pushes aside is the end of a lever (3); the other end is thereby moved out of the way of a bowling ball, which proceeds to roll down an inclined plane (5); having picked up considerable speed, the ball rolls off the end of the plane; it is prevented from damaging the floor by hitting the head of the teacher, which, having dreamed up such a stupid exercise, is plainly impervious to damage from a mere bowling ball.

Between thinking this up and writing it up, I read John's description of his invention. The similarity of final result is neither plagiarism nor accident; like stellar parallax, it's something that needed to be done. While his machine is more elegant mechanically, mine directly and more forcibly effects what his does only as an afterthought, and my patent application will be based on that.

The Bozo Filter

As Autodesk staff grew into the hundreds, it became increasingly apparent that it just wasn't possible to hire people in such numbers, especially in Marin County, California, without ending up with a substantial subpopulation of airheads, bozos, and crystal crackpots. This, combined with an electronic mail system which permitted any employee to broadcast mail to all others, worldwide, leads inevitably to "flame wars," interminable arguments which create a great deal of heat but shed little light upon the matter being discussed.

I define an airhead, based on decades of careful observation in Marin Country, the Home Nest, as an individual whose beliefs are independent of fact. Arguing rationally with an airhead is like trying to move your house by standing outside and pushing against a wall—it doesn't matter how hard you try or how long you keep it up: you get exhausted and the house stays put. With an airhead it goes like this. You're told something like "The calcium in quartz resonates with the calcium in your bones, healing back pain." Not wanting to get into details about the causes of back pain or the nature of the "resonance," you point out, "There isn't any calcium in quartz—just silicon and oxygen." By return E-mail, "Well, so what. It resonates anyway."

Since disconnection from facts is the Sign of the Airhead, I wondered if one mightn't be able to screen them out by asking a few factual questions. Only in jest, of course—I'm sure such a test would violate hundreds of laws in California, where chemical composition of urine is the only valid criterion for employment. I penned this on March 20, 1988.

While driving over the mountain at 15 MPH behind an idiot (almost always in a Chevrolet, if you're into cultural icons) who weaves into both lanes yet never manages to pull off on any of the 10 turnouts between Muir Beach and Tam Junction to allow the 10 to 50 cars behind him to get by, I often pass the time trying to imagine what passes for thought processes in these subminds—have they, for example, not really internalised the concept that there are other people in the world, but rather view the others around them as furniture or spear carriers?

This little diversion, which I usually undertake as a blood pressure reducing effort, of course, is often counter-productive because one then passes on to some of one's past and present co-workers... and some of the recent electronic mail communications and the worldview they represent.

Anyway, I was wondering how you could screen out the airheads: what is the essential distinction between a highly creative person who almost invariably seems flaky and the real crystal-powered airhead. Well, the essential difference, it seems to me, is that the real people are intensely interested in *facts* for their own inherent value and as the foundation of thought, and they understand things like one counterexample disposing of a global supposition.

So how do you tell the difference? In jest, I propose:

John Walker's Bozo Filter

1. What is the diameter of the Earth?
2. Who was President of the United States during World War I?
3. What is eleven times three thousand seven hundred eighteen?
4. What is the name of Autodesk, Inc.'s primary product?
5. What is the name of the largest computer company in the world?

Wouldn't it be interesting to see how many people who apply for work here could answer these questions, being generous for common errors (for example, for the first question if the person mistook the diameter for the radius, that's OK. "Duhhhhh" is not OK)? Similarly, in question 3, failure to propagate a carry is OK; not being able to turn the words into numbers isn't.

Wouldn't you love to know how many people working here today could pass this "test?"

I am not seriously proposing this for any purpose... it's just an idle diversion I'm sharing for your amusement.

From: Dan Drake, Mon Mar 21 09:09:36 1988
To: Chris Record, John Walker, Kern Sibbald
Subject: Bozo Filter

It sure would be fun to know.

My own thoughts on such subjects came when driving—an activity that seems to be favorable for meditations, most of a dyspeptic sort—on a freeway, watching the cars go at 65 mph separated by intervals of perhaps 15 feet: What kind of silly ass was C. P. Snow? He wanted people to understand the second law of thermodynamics when most of them don't believe in the conservation of momentum.

You may know my literacy test, which I don't dare administer to tech writers:

> "I hit him in the eye yesterday."

It's physically possible to insert "only" (well, or "Only") in 8 places in that sentence. For all cases that are meaningful, distinguish the meanings.

In May of 1988, Autodesk overflowed the office space available at the Marina Plaza complex we had occupied since 1985. We had to split the company between that office and a building at 3 Harbour Drive, further north in Sausalito. Recalling the problems we faced when the company was divided between offices in Mill Valley and Sausalito, Dan Drake suggested that we obtain free bicycles for inter-office transit, just like the Provos of Amsterdam. We couldn't get white bicycles, but we did get a fleet of black and pink clunker Schwinns. Not long thereafter, Kelvin overheard a disparaging comment about the bikes we'd bought. So, he was moved to write the following.

To: Uneasy Riders
From: Kelvin R. Throop
Subject: High performance bicycles
Date: May 10th, 1988

Grouse, grouse, grouse.

Not one company in a thousand would brave liability and flout convention to provide free Provo bicycles to permit primate-powered peregrination among the far-flung buildings of Autodesk's Sausalito headquarters.

But, of course, this is Marin County—Marvelous Marin, where the possible is bounded more by lack of imagination than constraints of reality and resources. So Autodesk's bicycle fleet is regarded with a jaundiced eye by the truly trendy, who say "a dérailleur is *de rigeur*!".

Indeed....

Because Autodesk believes so strongly in upholding the standards and image of Marin County, however mylar-thin and trivial, we have decided to solicit bids for high-performance bicycles to supplement the existing fleet. Turbo Digital Cyclery of Bolinas have agreed to screen entries and maintain the new bicycles after they are delivered.

The new bicycles will be equipped as follows:

Propulsion. Pedals adjustable for leg length from 0.5 to 1.5 metres. Kevlar belt and carbon-fibre cone microprocessor controlled continuously-variable transmission (CVT) delivering power to rear tire. Toe clips equipped with automatic impact-release mechanism.

Braking. Bendix carbon/carbon disc brakes on both front and rear wheels. Bosch computer controlled ABS antilock system with deceleration sensor balancing load between front and rear discs.

Guidance, Navigation, and Control. Martin-Marietta Lantirn Forward Looking InfraRed (FLIR) pod for night riding, presenting imagery in a helmet-mounted Head Up Display (HUD). Laser ring gyro inertial navigation system coupled to moving map display also presented in HUD, with optional superimposition with FLIR information. Backup coordinate fix system using LORAN, Navstar GPS, and Soviet GLONASS systems,

with automatic recalibration of inertial navigation data. Terrain and pothole database complete from the Bay Model[217] northward to Feng Nian.[218] Also a speedometer and an idiot light that indicates something failed.

Countermeasures. Computer controlled, expert system driven, automatic countermeasures suite. Automatic countermeasure delivery system capable of delivering Milk-Bones if chased by a dog (automatically sized to dog's jaw radius), chaff if illuminated by radar, and Lotto tickets if pursued by bozos. In addition, low observable techniques reduce the radar cross-section to less than 20 cm^2.

Survival, evasion, and escape. Zero-zero ejection seat, mortar-deployed quick-opening parachute, automatic inflating life vest with EPRIB transmitter and strobe light triggered by ejection. Watertight survival kit includes can opener, good-luck quartz crystal, PFIX 2.0, Oreo Big Stuff cookie, supply of requisition forms, and Torx screwdriver. In case suicide is required, a Sony Walkman and New Age music cassette are provided.

Fuel. Four 2 litre tanks are mounted below the centre of gravity. These supply, on demand, Jolt, Gatorade, Coiled Springs Mineral Water, and Diet Toxic Waste. An automatic crossfeed system maintains balance as well as delivering any desired mixture to the rider. The tanks can be jettisoned to improve acceleration in an emergency.

Weaponry. Few combat engagements are anticipated for these bicycles while fulfilling their inter-office mission. For those cases where there is no alternative, four Marinchip BGM-25L bozo-seeking missiles are mounted. These missiles home on the nearest erratically-steered, slowly-moving vehicle, then deploy a balloon in the shape of a Sausalito police car and emit the sound of a siren along with a speech-synthesised "Pull over, asshole", permitting the cyclist to pass safely.

Command and control. On-board cellular telephone, FAX machine, Quotron terminal, UPI newswire, and UUCP mail and news feed. The DIAL[219] system will be automatically alerted when the rider departs, so it can interrupt every trip with a "you have 35 messages" call, whether messages are waiting or not.

Configuration and performance. Curb weight (less rider) not to exceed 8,000 kg. Acceleration from 0 to 40 km/hour not to exceed 20 minutes with average rider. Training time to solo not to exceed 500 hours.

This is, after all, Autodesk. Why not the best?

[217] The Army Corps of Engineers hydrological model of San Francisco Bay and the rivers feeding it, located just south of the Autodesk Marina Plaza office.

[218] The Chinese restaurant behind Autodesk's first office in Sausalito. Much of Autodesk's strategy was plotted around a table at Feng Nian.

[219] Autodesk's infuriating voice mail system.

From: Carl Bethea
Date: 2-14-92 2:52pm
Subject: A Journey Not to Forget

(API) SAN FRANCISCO—Thousands of local residents crowded the Golden Gate Bridge as an unlikely convoy of buildings passed underneath. In a bizarre scenario that started three days ago, several buildings in Suasalito, a community north of San Francisco, became uprooted from their foundations during an historic rainfall and floated into the Bay. Autodesk president Al Green explained, "We made a decision to go for it. We could flounder in the Bay until we sank like a cast iron stock certificate, or we could head directly into the storm and fight for our lives."

Employees broke first-floor windows and began to paddle with whatever they could find. "As long as we all rowed in the same direction," explained a straight-faced Green, "I knew we could make it." When the Bay became too cluttered with other broken vessels, the convoy headed for open sea. Volker Kleinn, commanding a house boat, acted as tugboat to nudge and pull the buildings into safer waters.

"They could have wandered about the ocean for weeks just looking for each other," explained Malcolm Davies, who directed the rescue efforts from a hilltop in Marin—except for a brief period when the Juggernauts entered international waters. "Instead," he said as he leaned over a criss-crossed, sweat-soaked org chart, "they kept a tight formation, worked together, and had fun."

Upon docking again in Sausalito, a haggard Robert Wenig, leader of the ports group, hurried past reporters carrying a dog and saying only, "We will ship on time. Get out of the way." Green later explained that the dogs had been used in a jerry-rigged power generation system to keep work progressing on the Windows project of the company's flagship—er—popular product, AutoCAD, and that now the animals needed rest and food.

Asked how the three day sojuorn in the Pacific affected product support efforts, Lew Goldklang said, "What?" Green later explained that a microwave network link to the mainland had kept the castaways in constant contact with their colleagues. "We could have eaten those dogs," Goldklang said. Braced by an inventory of canned soda and vending machine snacks, most of the the suddenly sailors claim they never thought seriously about a food problem.

Meanwhile, other employees kept spirits high in a continuous party in San Rafael. "We knew they wouldn't want to miss it," explained Lisa McCormack, "so we just kept it going until they could join us." One employee was unhappy that she couldn't keep the 87 billion credits she had accumulated on the casino tables, and several local tuxedo rental firms were threatening to sue unless their tails were returned, but otherwise the company was none the worse for the adventure.

From: Jeff Wilson
Subject: Press Release
To: Headquarters

The following information was released this morning, April 1, at 9:00 a.m. PDT:

April 1, 1992
For Immediate Release
Contact: Annie Body
415/332-2344, extension 4400

AUTODESK ANNOUNCES BREAKTHROUGH PRODUCT

SAUSALITO, Calif. — In keeping with the new policy of putting "more in the box[220]," Autodesk today announced that it has successfully integrated its AutoCAD, AME, Stereolithography, HyperChem, Render, 3D Studio, Surfaces, CHAOS, and CyberSpace projects into a single design/modeling/visualization/construction program. The program, code named "AutoAdvanced3DSuperHyperCyberChemChaoticStereoSurfSpaceRendModExtension" (or simply "AutoThing"), is a complete design, modeling and construction system in which the user designs a model by merely visualizing the object in his/her mind. The object is then manufactured by Sterolithography Apparatus specially designed to handle large objects such as airplanes, oil tankers, and skyscrapers.

Ms. Bea Esser, spokeswoman for the company, stated, "The first milestone of the project was reached last week when, in a single 14 hour session, one of our developers was able to construct a cube". The developer, currently recuperating in a local Napa hospital, was unavailable for comment. She went on to say, "The project is on schedule and we can commit to a firm delivery date of sometime in the mid to late 21st century... maybe".

"We are working on the few remaining bugs," says the chief engineer, Mr. I. Chee-wowa. "The mental state of the user is key. For example, we have discovered that using the system immediately before lunch usually results in the output of Big Macs, fries, and the like. One user, a newlywed, had to be removed from the system after accidentally manufacturing quite a collection of... well, shall we say, erotic figurines." Autodesk feels these shortcomings can be overcome through proper training.

The program also includes a new form of copy protection in which the user is mildly brain-damaged if the software is not properly registered. The project suffered a minor setback when one user was fatally brain-damaged when the program crashed. The company has indicated that they do not consider this a "stop ship" bug and will address the problem by including a free life insurance policy with each copy.

"This is a breakthrough product because of the manpower savings it represents in the design and construction trades.", said Mr. Duit Tuit, vice-president of Marketing for Autodesk. "With strong sales we feel we can get the unemployment rate up into the 40 to 50 percent range. Homes can be constructed in a matter of hours." Mr. Tuit also mentioned that some reconstruction may be necessary in the spring because objects created with the new program tend to dissolve during the rainy season.

No price has been set for the new product, but company officials are vowing to keep it under 7 figures thereby upholding the tradition of providing 80 percent of the functionality for one tenth the price. In keeping with its

[220] A reference to Information Letter 14, released precisely one year before. See page 606.

commitment to its current customers, Autodesk also announced that full upward compatibility with previous versions of AutoCAD will be maintained. The new program will initially ship on Cray supercomputers, but a DOS 286 version is expected in the quarter following its initial release.

Autodesk develops, markets, and supports a family of computer-aided design, engineering and animation software products for April Fools like you.

Flat-out Programming

In August of 1986 I began the transition to full-time programming. This was the memo that announced that I'd be concentrating primarily on software development for three days a week. This was the first step in my plan to get out the job of president and work, instead, on things I was good at.

To: Everybody
From: John Walker
Date: 26 August 1986

Subject: Flat-out programming

This company has committed itself to the release of a 3D version of AutoCAD by the end of the year, to shipping AutoSketch in the month of August (our failure to meet this promise is now a foregone conclusion), to shipping the MicroVAX by the end of next January, not to mention numerous other clamant technical projects such as AutoShade, the mechanical template, Apple machine support, a better user interface for AutoCAD, networking, and performance improvement.

Making these commitments was not grasping for the sky; they are central to the continued survival and growth of our company, for only by building a diversified product and machine base can we insulate ourselves from competitive onslaughts and the vicissitudes of the market in general.

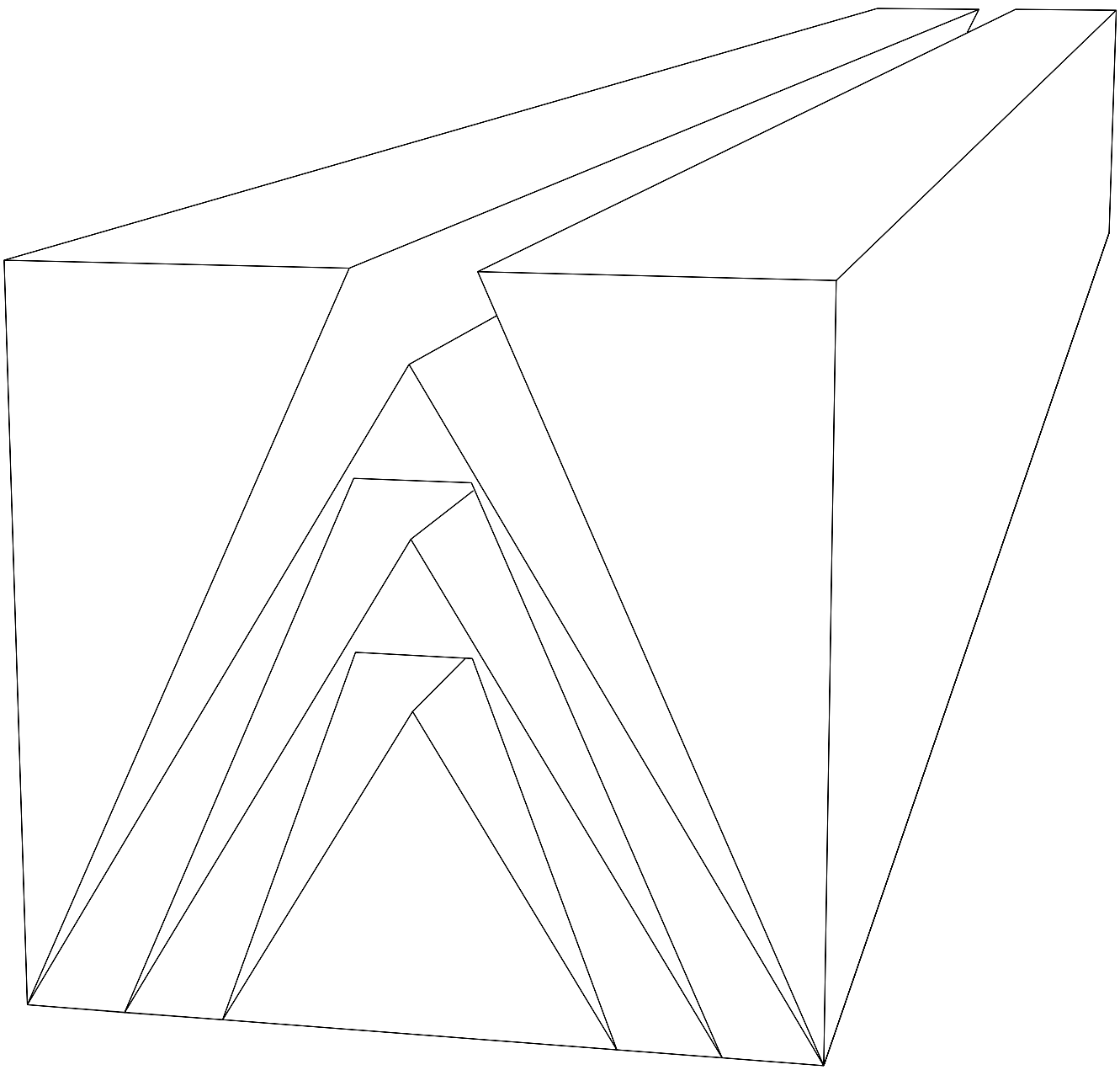
We will fail to achieve our goals if we squander the limited resources we have on non-vital tasks. Consequently, I have decided to focus my time and energy far more on software development than I have done since February. To this end, and starting tomorrow, I will be working from home on Tuesdays, Wednesdays, and Thursdays. I will continue this until we ship a 3D AutoCAD to the first paying customer, and until the backlog of critical technical projects is cleared.

Any items which would have been brought to my attention during this time should be referred to Dan Drake, who has complete authority to act when I am out of the office.

This is not a mandate or invitation to stack up Mondays and Fridays with meetings with outsiders who "have to talk to the president" or can "only deal with the CEO". I cannot recall a single significant statement uttered by one of these blithering pithecanthropoid bimbos which I needed to hear. If they can't tell it to somebody in the company who can really help them, it will not be heard. I will deal with such requests in my normal courteous manner.

"What does this mean for the direction of the company?" Oh come on, now! During the times of this company's most rapid growth I was primarily spending my time contributing to the technical excellence of the products

on which our success is founded. If we fail to be the technological and performance leader, we will fail in the market. It is time to take every step to prevent that.



Hardware Lock Debater's Guide

Rhetoric is a much maligned and neglected skill in this inarticulate age. The introduction of the hardware lock in AutoCAD version 2.5 afforded a superb opportunity for Autodesk folk to hone their debating skills; a great deal of energy and large chunks of 1986 disappeared into The Great Hardware Lock Debate. Intense technical, ethical, and philosophical arguments swirled within the company, our dealer network, our customers, and the software industry itself. Emotions ran high, as most participants held opinions at extreme ends of the spectrum. At the height of the debate, Product Support drafted this position paper, explaining our reasoning. We decided to discontinue the lock on domestic products anyway, in recognition of the deep cultural chasm between customers in North America and the rest of the world. In retrospect, it was a sound business decision, as we defused most of the antagonism directed against us, with no apparent loss in revenue.

In Defense of the Hardware Lock

September 23, 1986

By Victor Zlobotsky, Rear Admiral USN (Ret.), visionary inventor of the superheterodyne crystal and general-purpose energy transponder, technological precursors to such labor-saving devices as the Norden bombsight and the programmable microwave oven.[221]

The Law is on our side: The courts have held, correctly in our view, that software is both copyrightable and patentable; that "works in code are quite privileged from an intellectual property perspective." Running unpaid-for copies of AutoCAD in violation of the Software License Agreement, is in fact, theft, even though copying disk files is temptingly trivial. The hardware lock is the simplest way for us to protect our legal rights in safeguarding our property from unauthorized use.

Protection of License Agreement: Autodesk is committed to maintaining the integrity of our Software License Agreement. The hardware lock is the most painless, least obtrusive means of assuring compliance with the Agreement; it supplants such awkward artifacts as special master diskettes and hard disk installation counters.

It is <u>not</u> copy protection: It is Software License Agreement protection. The hardware lock allows unlimited copying of AutoCAD software files for backup/archival purposes. Users can make backup copies, load AutoCAD onto hard disks, run from network file servers, and have copies of the program installed on multiple machines. Only a device like the hardware lock allows such flexibility and freedom of file duplication while enforcing the License Agreement.

[221] Admiral Zlobotsky is, for what it's worth, a spitting image of Lew Goldklang, who wrote the introduction to this chapter.

Fair compensation: By protecting the License Agreement, we protect our profit margins, which helps us grow quickly enough to meet the explosive demand for AutoCAD and related products. The features incorporated into Rev. 2.5 and products still on the (electronic) drawing board, all demanded by a wildly enthusiastic user community, are made possible by our ability to grow and add talented individuals to our enterprise. This also allows us to increase the staff required to provide product support to a rapidly growing base of over 60,000 installed users.

Protects users' investments: The lock protects legitimate users from the unfair competitive advantage exacted by dishonest users who would like to run unauthorized copies, yet still draw on Autodesk's resources, like phone support. With the lock, licensed users who pay for AutoCAD receive the support they are entitled to. Software pirates also receive what they pay for: zero. The hardware lock promotes a "level playing field," assuring that licensed users are not subsidizing bootleggers.

Realistic pricing: By stopping the proliferation of unauthorized copies, the lock allows us to price AutoCAD fairly. Widespread software piracy otherwise tends to cause price inflation to recoup revenue lost to illegal freebies.

Wide user acceptance: The lock has enjoyed wide acceptance among our customers, the overwhelming majority of whom are scrupulously honest and appreciate the importance of protecting their License Agreement in this way. Since its introduction with the release of AutoCAD 2.5, which started shipping on July 8, less than 15% of all calls into Product Support have been complaints against the lock.

High reliability/low failure rate: Out of 13,400 locks shipped, only 38 have been returned as dysfunctional, a failure rate of only 0.28%.

Still a bargain: Autodesk has pioneered in bringing CAD technology to a wide audience. We have exploited the recent dramatic advances in micro-based computing power and graphics technology, slashing the cost of a richly functional CAD system by a full order of magnitude. A fully-configured AutoCAD system, providing over 90% of the functions that as recently as a few years ago were available only on a mainframe at a cost exceeding $100,000, is now available for less than $10,000. AutoCAD remains the most sophisticated, popular CAD software on the market at any price. Insisting that customers pay for legitimate copies is a modest, reasonable business decision.

Rebuttals To Frequent Complaints

Site Licensing: Autodesk should drop the offensive lock and offer site licensing instead.
Answer: We have chosen a policy of quantity discounts for multiple purchases, rather than offering site licensing. Substantial price discounts are available to users with multiple workstations. It is curious that interest in site licensing has jumped remarkably since the introduction of the hardware lock, implying a considerable number of users running the package in violation of the Software License Agreement.

User-vicious: The lock is a user-hostile device that is a royal pain to install.
Answer: *Au contraire!* The lock is easy to install; certainly no more difficult than hooking up any other peripheral device.

Interferes with other programs or plotter: Since the lock takes up my **`COM1`** port and interferes with some of my peripheral devices, (they cannot be attached to the lock on the **`COM`** port directly), Autodesk is forcing me to buy another board just to have a second serial port. This is an unanticipated additional expense, and a major recabling headache.

Answer: The lock actively drives certain pins in the **COM** port; it is not a purely passive monitoring device. It is transparent to most other application programs, such as word smashers and spreadsheets, but it is not transparent to a limited class of devices, such as streaming mode digitizers and parasitic (power sucking) mice. There is no simple work-around for this.

No COM1 port: I can't run (or upgrade to) Rev. 2.5, since my machine does not have a **COM1** port at all.
Answer: This is a degenerate case of the previous interference complaint. The number of users affected is statistically very small.

Backsliding/unraveling: My system worked fine with the previous revision of AutoCAD, but the hardware lock refuses to work with my existing cabling, causing unwarranted replacement expense.
Answer: The hardware lock is somewhat more exacting in requiring that all connecting cables be up to the standards specified in the Installation Guide. Users who jury-rigged quick cables on the cheap, like the old 3-wire plotter trick, will have to bring their cabling up to reasonable electrical standards. In these cases, the hardware lock is helping users by eliminating faulty cables that may pose electrical and fire hazards.[222]

Lock is a "weak link" in system: It's an intolerable nuisance to have my entire business/office/drafting system depend on a fragile piece of plastic that is easily lost or stolen.
Answer: The lock is an integral part of the system. The same proper caution and security you use to protect your computer should be extended to include the lock. You can't drive your car without your car keys, either, yet we all live with them and exercise sufficient care for their protection. Authorized resellers may request a spare lock for their inventory. There is also a reasonable procedure to replace locks quickly that are found to be defective in the field.

Over a year's experience in Europe has shown that a system is far more likely to be brought to a stop by a bad keyboard or hard disk than by a lock failure. Note also that a lock failure, if it does occur, does not cause loss of work-in-progress.

Susceptible to theft: The lock is an external appendage that is too easily stolen by mischievous hooligans.
Answer: We are considering offering an internal lock that would tuck inside the machine's "skin" for certain popular computers. Since we support so many different machines (over 70 at last count), the wide variety of individual computer design and packaging make the production of multiple, physically different internal lock designs prohibitively expensive.

9-to-25 pin adapter: Autodesk should ship 9-to-25 pin adapters to those users who are now forced to "roll their own" as a result of the hardware lock.
Answer (under investigation): We are negotiating with several suppliers who can provide high-quality connectors at a reasonable price. We are investigating the possibility of including these adapters on certain versions of AutoCAD (like the IBM AT).[223]

Low marginal production costs: Autodesk is ripping me off by charging outrageous sums for installing additional AutoCAD workstations in my shop, when their marginal production costs are so low. Each AutoCAD unit can't possibly cost more than 59 cents for the floppy disks, plus a few pennies more for the packaging.
Answer: Our marginal production costs are actually quite high, although it is less obvious from the physical appearance of the final product. Hardware manufacturers, by contrast, crank out tangible, heavy units where the cost of the raw materials and metal-bending is far more visible. Our costs are labor-intensive. We have to

[222] You may think this extreme. In twenty-one years in the computer business I have witnessed four computer fires and two close calls.
[223] Oh, indeed we were. We had zillions of 9-to-25 adaptor cable sets made up, for free distribution. They arrived just about the time we decided to discontinue the hardware lock.

pay all the salaries for software development, marketing, sales, product support and training. We pour more than 20 man-years per year into the development and refinement of AutoCAD, continually adding new features requested by our customers.

Swimming upstream: Autodesk is foolishly bucking the clear industry trend by protecting its product at a time when most other vendors are abandoning protection schemes of every stripe.
Answer: Untrue by virtue of false comparison. Most competitive software costing more than $2,000 is protected, including products from CalComp[224], Computervision[225], and McDonnell-Douglas[226], all of which use a hardware protection device like ours. Some publishers of mass market software are dropping copy protection because of concerns about user inconvenience with limited numbers of master disks, hard disk install and de-install, and hard disk backup—problems that don't arise with a hardware device.

Invasion of privacy: The hardware lock is fundamentally un-American; it violates my right to privacy under the first, fifth, eighth and eighteenth amendments to the Constitution, and my lawyer will be serving you with a subpoena next Tuesday.
Answer: The lock is no more an invasion of privacy than any other security device. It's there for your long-term protection, including the safeguard against one of your own employees violating the Software License Agreement unbeknownst to you. In corporate environments and small, multi-person offices, the lock actually eliminates the need for employers to play the unpleasant role of software police.

Philosophical self-righteousness: "There should be no secrets between any two sentient beings in the universe", and this logically extends, by induction, to the moral repugnance of copy-protecting any software.
Answer: Timothy Leary expounded this belief once, and now he is a software publisher himself, carefully watching his own margins. (Yes, but is Mind Explorer copy protected?)

Hard-core intransigence: I don't buy any of these arguments, and I'm so incensed at the hardware lock that I will purchase all future CAD software from a competing publisher, just for spite.
Answer: If you are that adamant about the lock, try using AutoSketch. It runs without the lock, is inexpensive, yet gives you over 75% of AutoCAD's functionality.

[224] Discontinued
[225] Acquired by Prime
[226] Discontinued.

Information Letter 13

Information Letter 13 was distributed concurrent with the announcement that I was relinquishing the presidency of Autodesk and turning the office over to Al Green. This letter tried to convey the incremental nature of the transition and to focus people on the challenges that lay ahead.

Autodesk, Inc.
Information Letter # 13

by John Walker
Revision 7 — November 5, 1986

"... we have tried the utmost of our friends, Our legions are brim-full, our cause is ripe: The enemy increaseth every day; We, at the height, are ready to decline. There is a tide in the affairs of men Which taken at the flood leads on to fortune; Omitted, all the voyage of their life Is bound in shallows and miseries. On such a full sea we are now afloat, And we must take the current when it serves, Or lose our ventures."

Shakespeare, *Julius Cæsar*, Act IV, Scene 3

What a long, strange trip it's been.

It occurs to me that more than half of the people who work for Autodesk have never had the experience of having one of the these rambling Information Letters plop into their mailbox. For those of you reading your first Information Letter, I'll just say that these Letters were the primary means of communication in the early days when we were trying to get the company together. For the first year, we actually only all got together about once a month, so the Letters served to pass information around economically and force us to put on paper, in specific form, what might be only a mumble in a meeting or on the phone.

It's been a long time since the last Information Letter, and since our recent management reorganisation might leave some people wondering just what is going on around here, I thought I'd put electrons to silicon and bring everybody up to date. Please don't attach any significance to the number Thirteen; that's just the next one in the series.

Ground rules

I've always tried to be open, up-front, and straightforward in these Information Letters. I'll continue that herein. I am not aware of any euphemisms, dissembling, or "cover stories" in any of the material in this Letter. Since some feel moved to weave intricate stories of intrigue around any change that occurs, such protestations may be wasted, but if you've come to believe what I say, believe me when I tell you that this is the straight stuff. The only punches I pull are to honour nondisclosure agreements with manufacturers regarding projects underway, and new product strategy information I don't want to make publicly available.

The management switch

Back when we were thinking about getting venture capital, I experienced the joy of having strangers walk into a room with me, talk for about a half hour, and then be told by them that I wasn't a "strong CEO" and lacked a "track record" because I hadn't "done it before". It may be one thing to run a small struggling operation but, they said, entirely another when your sales went to $1 million a month and you had a "real company here".

I responded that I felt I was entirely competent to run the company up to the ten million dollar a year level, and that after that I thought I was smart enough to know when I began to get out of my depth. I assured them that my ego wouldn't blind me to what was best for the company and I would gladly hand over the company to anybody who could do it better.

In February of 1986 I began to feel that the time might be coming when I should re-examine continuing as president of the company. At the time, I was concerned about whether I had the skills to build the kind of organisation we would need to reach fifty million, then one hundred million, and more in sales. I thought about this long and hard, had numerous conversations with various people in the company, and concluded at the time that I might as well keep on slogging away at it.

Now's where I'm supposed to say that the light shone upon me, and in October I woke up saying, "*now is the time*". Well, it didn't happen exactly like that. . . . Cause what it comes down to is I think I could continue to run the company pretty well for as long as I could stand it before blowing out physically or mentally, but that my being president is not in the best interests of the company. And here's why.

Look, for most of the history of the company, I was spending 8 to 12 hours a day on product development, including programming, talking with others about design issues, meeting with vendors and users, writing manual inserts and ad copy, and so on. I spent the remaining 4 to 8 hours being president. Now for all of that time I was programming I was *neglecting the job of president*. The president is supposed to be the company's interface to the outside world, representing the company to investors, analysts, giving speeches, and so on, and of course "running the company" on the inside: monitoring progress, watching financials, and planning the development of the organisation. We were very fortunate in having such a large group of founders with such diverse talents, and we were equally fortunate in being able to recruit equally talented and extremely dedicated people in the early days who looked at the work to be done, set to it, and got it done with little or no supervision. So the company required very little explicit "running", and consequently my concentration on the technical front did not lead to the catastrophe one would expect in the classic Silicon Valley company where only one or two founders have the dream and the rest are working for salary.

I did not come to this odd division of time entirely because I enjoy programming more than being president

(though I would be less than candid to say that had nothing to do with it). I did it because I felt that it was in the best interests of the company; a criterion by which I hope I have evaluated every work-related decision I have made in my career. Let me continue with candour, even at the cost of humility. I believe that I am one of the most productive programmers presently living on Planet Earth. I think that in 19 years of programming, I have adequately demonstrated I can go from concept to product in less time than most others, even those considered highly productive, and in conjunction with a small team of others as competent and highly motivated as I, can take on and complete tasks normally measured in calendar years and tens of man years. This is not bragging; it's a statement of what I believe and since it plays a large part in my recent decision, I want to share it with you so you can understand my reasoning.

The very things I consider I do poorly are the things that largely make up the job of president. Worrying about building space, staffing plans, budget versus actuals, day-to-day project status, contract terms, and the like is vitally important—but those are things that I'm not very good at and frankly am not much interested in. They tell me I do a pretty good job at giving speeches, but the tension and effort puts me out of commission for about 4 days before and 1 day after each one—and I don't think that's a good use of what I have to contribute to the company. All of these things were, in the not too distant past, matters one could attend to in less than 10 hours a week, but as the company has grown, they've grown to that horror of horrors, a *full time job*. And I'm a poor candidate to fill it.

When I decided back in August to spend more time programming and less time "in the office", it was in part to see how well the company would run with Al Green and Dan Drake filling in for me. It worked superbly. Not only was I able to complete AutoSketch and undertake another project that will be revealed at COMDEX, the overall operation of the company improved (as I expected). In addition, I managed to lose 10 kilograms and not go crazier than usual.

So, the test having succeeded, now's the time to make it official. Now, because doing it before COMDEX will let us explain it in person to our dealers, developers, and users at COMDEX. Now, because by doing it at the end of the fiscal quarter, we can communicate it in our quarterly report to the financial community. Now, because Al Green and Dan Drake have been doing the job for the last four months (and in a large part for the last year or more) and deserve the formal recognition for the job they've excelled at. Now, because the technical and competitive challenges that face us are unprecedented, and we can ill-afford to misapply my talents to administration and speechifying.

> "You have sat too long here for any good you have been doing. Depart, I say, and let us have done with you. In the name of God, go!"
>
> Oliver Cromwell

All right, all right…I'll come to the point. We have talked for years now about how the rapid advances in computer technology will inevitably bring the power of the mainframe to every person on this globe who seriously wants it. We endured the derision of those who told us "you can't do CAD on a PC", and now we listen to them explain how they predicted it all along. Every major CAD vendor now competes with us in the PC market, and each now concedes that our core market, 2D drafting, is best served by a PC-based product. As the power of the PC grows, a new challenge faces Autodesk, and we'd better be as far in advance of this trend as we were in 1982 with AutoCAD. In putting AutoCAD on a PC, we were applying our skills as programmers in shoehorning a minicomputer program onto a machine one tenth the size. With the advent of the 68020 and 80386-based workstations this year, the distinction between PCs and mainframes has been erased. Please reread that last sentence two times. If you agree with me that it is true, it has monumental implications for the survival and future of this company.

"There is no avoiding war; it can only be postponed to the advantage of others"

Niccolo Machiavelli, *The Prince*

What it means is that we're moving from a contest of who can cram the most features onto a desktop to a question of who can provide the best tool, period. Most of the constraints of memory, processor speed, secondary storage, and graphics resolution that we assumed in the 1960's and 1970's would hound us until our retirement have been erased by Man's greatest triumph of mass production: photolithography on silicon. The arena now opening and which will probably occupy the next decade is: who can provide the tools which best aid a creative person in turning ideas into reality? I urge you to forget all of that MBA-bullshit about market segmentation, channels of distribution, end-users, value-added, strategic positioning, and the like. I suggest that presuming competent marketing, sales, and distribution, the company that best solves the problems that face designers will end up with the market. Forget PCs versus workstations versus mainframes—that's history. We're building tools, and the tools which work best will endure regardless of the materials of which they're made.

"Business-minded decision-makers must learn to invest precious, high-value, near-year dollars to recoup discounted out-year operations and maintenance savings."

George M. Hess, Colonel, USAF

Right. It comes down to *reality*. Imagine the concept. What we've been doing for the last four years is building a tool which embodies a language—drawings—which we use to represent reality. This is a language which goes back thousands of years (re-read the *Scientific American* article about the guide-lines used to build the Parthenon), which represents artifacts in a compact and unambiguous form. AutoCAD is one helluva drawing tool. That's step 1. From here on in it gets harder. When we move from AutoCAD as it exists today to a true 3D AutoCAD, we take the first step on the road to modeling—embodying a physical system in the computer. Once you've encoded a model, you can do many things with it—calculate physical properties, generate part programs to build it, ask "what if" design questions, and so on. I think that every practitioner in the modeling field agrees that the tools we have today are stone axes and bearskins compared to the tools which will evolve over the next ten years. The tools we struggle with today all date from the era of vastly expensive computers with limited memory and poor graphics, and they will be consigned to the sandbox of technology as the new generation of tools appear.

But these tools do not "evolve". They do not "appear". They are built by the mental exertions of hardworking men and women, competing in a marketplace that winnows the ill-conceived and inefficient and bestows incalculable rewards upon those who meet the challenge. Autodesk is today a central figure in this competition—what we've done so far has placed us there. The products we develop in the next five years will determine whether we become the force that liberates the minds of designers from the tyranny of calculation and delivers the power tools of modeling from the few to the inventive multitudes, or whether we gather dust in the archives as a footnote in the article on "Design—computers—1980's".

Spock: He is intelligent but not experienced. His pattern indicates two dimensional thinking.

Kirk: Sulu... translate Z minus ten thousand.

Star Trek II, The Wrath of Khan

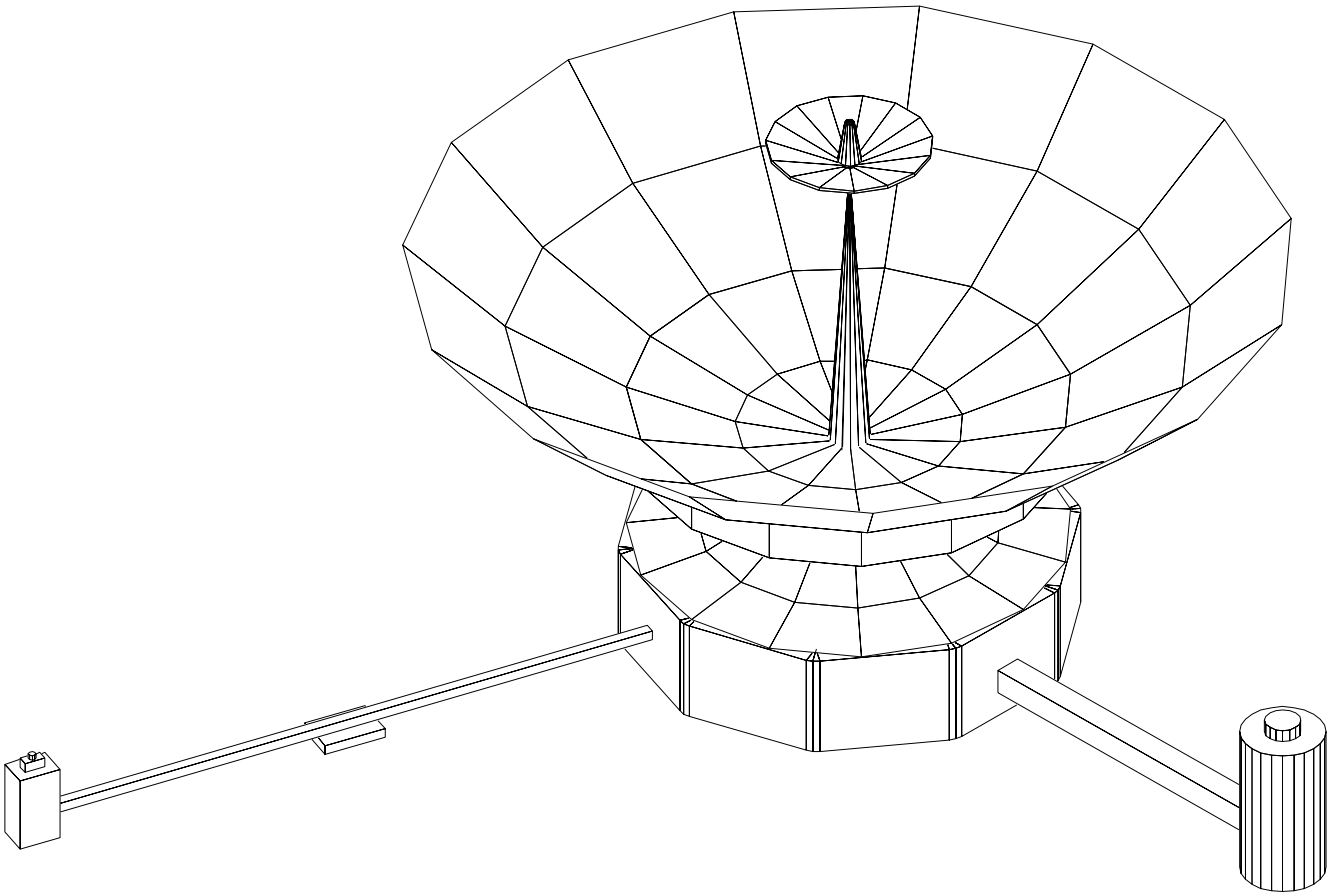
So now we undertake to put, piece by piece, the whole wide world into that little bitty can. We start with a 3D AutoCAD. We add a little thing I've been working on. We labour, patiently and indefatigably, to add the pieces until a designer suddenly discovers a new way of working—a way that's as much an improvement on the old as CAD was over the drawing board or a word processor over retyping pages. I believe in tinkering. I believe in the individual as the source of ideas and inspiration. I believe in providing that individual with tools to explore his or her ideas as powerful as those available to the Pentagon or the Politburo. I believe that the inevitable development of technology will eliminate the advantage of mere size and restore creativity and innovation to the respect due it and paid it through most of the history of this country. I believe that those who make those tools will earn rewards, financial and moral, which will make the wealth generated by Autodesk to date appear insignificant. I believe that Autodesk stands today as the clear leader in the quest to develop these tools.

> "We shall see how the counsels of prudence and restraint may become the prime agents of mortal danger; how the middle course adopted from desires for safety and a quiet life may be found to lead direct to the bull's eye of disaster."
>
> Winston Churchill, 1948

Those who know the scope of what I'm talking about may shrink from the magnitude of the job to be done. Creating a realistic picture of a forest at sunset requires more software development than has gone into AutoCAD to date, and more computer time than has been consumed by every AutoCAD user so far, worldwide. Modeling the forces on a tire rolling through a pothole exceeds the capabilities of any existing hardware and software. Increasingly, we will have to build systems of extraordinary complexity which work the first time. Five years ago, Autodesk was a vague idea circulating in my head classified as "new software company". Our collective efforts will determine, five years from now, whether our progress continues to astound those who assume that competence, commitment, and candour must always be bested by money, management, and marketing.

I am proud of what we've accomplished so far. I am proud of what we're doing. And I'm proud to continue to contribute to our company those things that I do best—full time.

I made the Voyager drawing the night before leaving for COMDEX in 1986 where we were introducing AutoShade. It was based on a picture of Voyager 2 in Scientific American.



Cadetron and Solid Modeling

By mid-1986 there was little disagreement with the assertion that solid modeling would be a key component of the company's future product mix. But as we studied the requirements of the mechanical parts design market, which presently makes up the largest part of this market, it was also clear that even if AutoCAD could be extended into some kind of solid modeler, it would not really meet the needs of that market. Eric Lyons was the leader in researching this market, and in this paper he recommended that we investigate purchasing Cadetron which, in March of 1987, we did. As AutoSolid becomes more and more central to our strategy in the mechanical engineering market and begins to contribute a growing component of Autodesk's revenues, it will be fulfilling the promise that this paper evokes.[227]

How Autodesk Can Take Over the CAD/CAM Industry

Eric Lyons
11/21/86

We stand at a crossroads. We are about to be dragged—kicking and screaming—into a world we know little about: the world of engineering modeling. This, in itself, poses little significant threat. After all, no one knew anything about drafting when this company started out. But we had an advantage—we were the first to do it [on a PC]. So there was some leeway, some time to make up for the features we lacked. We no longer have

[227] *Note added to the Fourth Edition.* Alas, AutoSolid was never to become "a growing component of Autodesk's revenues." Launched in June 1988 as a stand-alone product which ran only on SCO Xenix (a platform upon which, at the time, AutoCAD did not run), at $5000 a copy, it sold very poorly. In 1989, I launched "The Eagle Project" with the goal of demonstrating AutoSolid's functionality integrated into AutoCAD, taking advantage of the ADS and Extended Entity Data features then being developed for Release 11. This was demonstrated, on schedule, by July 20, 1989 (the twentieth anniversary of the first Moon landing). Subsequently Autodesk decided to develop my original prototype into a genuine integration of AutoCAD and AutoSolid, priced at $500. This was launched (almost silently, see page 618) along with Release 11 in October 1990. In the process, the Atlanta office of Cadetron was closed, with some employees being relocated to California and others let go. This inevitably caused morale problems (see page 501). Subsequently, Autodesk licensed ACIS and began development of a successor to AME based upon that modeling technology, replacing the PADL modeler used in AutoSolid and AME.

In October 1992, Autodesk acquired Micro Engineering Solutions (MES) Inc. of Novi, Michigan, developers of the Solution 3000 CAD/CAM software, and announced plans to integrate this technology into AutoCAD. Work on AME was stopped at that time.

In November 1993, Autodesk simultaneously introduced AutoSurf Release 2, the integration of the MES modeler into AutoCAD using AME-like technology, and the acquisition of Woodbourne, Inc., formerly an AutoCAD developer, and plans to launch its modeler under the name of AutoCAD designer. As of late 1993, Autodesk has yet to "Take Over the CAD/CAM Industry" by establishing itself as the dominant force in solid modeling. But we're still working toward that goal.

Eric Lyons, the author of this piece, left Autodesk in September 1991 to found, along with Alvy Ray Smith, Altamira Software Corporation, developers of Altamira Composer. Autodesk was an initial equity investor in Altamira.

that luxury. Companies—big companies—that know and understand engineering modeling have seen that the PC isn't a toy; indeed, it and its successors are becoming the platforms on which their products are designed to run. Yikes.

We underestimated the importance of modeling over a year ago. We have been working on a generalized 3D version of AutoCAD for some time. In my "3D design paper" of May 27, I described 20 features that would bring us to a level equivalent to our competition at that time. We have implemented only a few; the hard ones are yet to be even started. And we have added things that never appeared on the list—primitive solid objects that are unrelated to each other. So we still have some work to do. But in the six months since I wrote that paper, some trends have emerged.

I have seen solids modeling, and it is the future. For years the skeptics have criticized solids as impractical, compute intensive, and inflexible. "You can't cut chips with solids," they'd say, or "sure they're fun, but what can you do with the model when you're done"? Well, they were wrong. Solids modeling is the next step in the evolution of CAD/CAM as surely as 3D followed 2D. Perhaps not for the better, but inevitable nonetheless. So we are faced with another problem: will we let solids get away from us as 3D did? Will we be sitting here in another year, wishing we had a solid modeler, and being run over by our competitors as we try desperately to catch up?

First of all, why is solids modeling such a big deal? See the attached article that describes some of its advantages over the more traditional surface modeling systems. Suffice it to say that its greatest advantage is that it is nearly impossible to create unmanufacturable objects with solids. Not completely impossible, but much less possible. In addition, analysis of solids is much easier and more accurate than with surfaces. So that's the big advantage. Solid modelers do require a fair amount of compute power and storage, however, so they aren't so great for AEC applications (a building described as a set of solids—including all of its components—would quickly be unmanageable on any of today's computers), but for discrete manufacturers they can be a boon. And there are a lot of discrete manufacturers out there.

The conclusion to be drawn here is that we need a solid modeler, sold by Autodesk, interfaced to AutoCAD, within a year. I rule out the possibility of making it part of AutoCAD within that time; 3D has taught us enough of that. Besides, solids is still frequently thought of as an application, and therefore the concept of a solids package outside of AutoCAD is fairly easy to sell.

So, what should a solids package do? Obviously, there is lots of room for definition—you can buy a $50 solids package for the Atari, or you can buy a $100,000 Medusa for the VAX. What's in between? First, consider that solids is a *design* application. Designers are interested in how objects behave. So beyond the ability to create an accurate geometric model using CSG or B-REP techniques, the thing is useless without being able to derive mass property information from the model. And since a large portion of engineering is spent determining the effects of loads, strains, and temperature variations on parts, the model should be efficiently interfaced to a FEM package and mesh generator. Also, since 80% of all parts manufactured in the U.S. are still done from engineering drawings, there should be some way to detail the finished part with a drafting package (AutoCAD is a decent choice). Ideally, this detailing step should be augmented with numerically controlled machine tool program simulation and verification for automatic manufacturing. And, of course, the user should be able to visualize the model in a realistic form. So that's what a solids package should do. Some companies have based their products on all or part of this definition. Aries Technology has spent 2 years and $15 million to bring a system to market that does solids, material properties, and FEM interfacing. A whole industry is spawning that the analysts call MCAE—mechanical computer aided engineering. Within a year, we will be competing with these people for the middle range of CAD/CAM.

Okay, we need a solids system. How does that affect our current developments in 3D? Well, we should obviously finish the development of 3D AutoCAD. With the exception of the funny little solid primitives we have defined, what we are working on is really a 3D drafting system. It is a way to define a wireframe model, view it, and detail it from any orientation. This is a necessary function when we *do* have a solids package, as well as being required for AEC applications. Where we go from there (surfaces, properties, etc.) depends on some decisions we make with respect to a modeler.

How Do We Do It?

We have two alternatives:

1. We write a solids package ourselves.
2. We buy one from somebody.

If we choose option 1, I don't believe we can acquire the knowledge necessary to write one from scratch in the time we must, so we must get a head start from somewhere. One possibility is to buy a one-time PADL–2 license from the Production Automation Project—80k lines of Fortran (FLEX, actually), designed to run on a VAX, $50,000—and convert it into something usable. Another possibility is to license the geometry libraries available from Applied Geometry. These libraries represent, perhaps, 15% of a working solids package. Just add code (*à la* Visual Engineering[228]).

A third possibility is a buyout of a company who has already done all this stuff, have them interface their product to AutoCAD, and sell it as our solids package. That company is Cadetron. Attached is some information on their company and their product. What follows is a proposal for the acquisition of their company and their product, and how it fits into our future: taking over this industry once again.

Product Positioning

Given that we acquire this product, how do we fit it into our existing product line, and what problems does it solve for us? First of all, the modeler would exist as a separate product, working as a pre-processor to AutoCAD (more accurately, AutoCAD would be a post-processor to it). It is intended for the mechanical engineering/manufacturing market. Parts can be designed, analyzed and realistically visualized using this package, then sent to AutoCAD (as either a fully 3D model for future 3D, or as a 2D "drawing" file) for detailing. The model can also be interfaced to FEA programs by using the optional automatic mesh generator. Eventually (they currently have this stuff under development), NC toolpath verification and simulation can be done on the model, and surface modeling (for things like car bodies and thin shells) can be done on the modeling side. AutoCAD is used only for detailing, in both 2D and 3D.

[228]The original concept of AutoShade was an interface to a library of rendering software from a company called Visual Engineering. Visual Engineering was never able to make their code work acceptably under MS-DOS, and we abandoned that effort and wrote AutoShade from scratch in-house. This comment is a reference to the horrors we encountered in "just building on an off-the-shelf library".

Also, AutoCAD 3D is used for virtually all AEC applications. Facilities management, piping, architecture, etc, are all done using the wire frame modeling in AutoCAD. We do not invest in adding surface modeling capabilities to AutoCAD, nor do we make our funny little solid primitives into a fully general solids system. AutoShade is still the AutoCAD rendering package, and is used by people who don't want to do mechanical solids applications. If you are an architect, you buy AutoCAD (and AutoCAD AEC stuff). If you are a discrete manufacturer (you make parts), you buy SolidWorks (or whatever we want to call it[229]) and its add-ons, along with AutoCAD for detailing. Perhaps we could even use AutoCAD as a front end to the solids package: we can create our funny little solids, rotate them, scale them, position them, then pass them on to SolidWorks for Booleans and analysis.

Eventually, the two products converge into one. They share a common user interface and a common database.

With this combination—the industry standard drafting software, the world's most powerful solids modeling software—all sold by the world's lowest cost CAD distribution network, people like Aries don't stand a chance. They will be forced out of the market. Computervision will shake in their already soggy boots. Intergraph will die a slow, horrible death, buried in caskets made of Interpro 32C boxes. Autodesk will prevail as the dominant force in CAD/CAM worldwide.

How Does All This Really Work?

Obviously, there are some things to be worked out. How much do they cost, how does the interface work, what modifications do they need to do to their product, how do we resolve the different operating system problem, making them understand what *really* low cost software should be, etc. A total buyout seems like the best opportunity to control all the marketing (especially) and the development direction. But I think it's important that we don't screw up a good thing—they have a team of 6 programmers who have successfully converted some of the most sophisticated engineering code in existence (while considerably optimizing it) to a small machine. Us telling them what to do with solids would be like them telling us how **DIMZIN** should work. We need to look at their commitments, review their development projects, and define a set of projects that will result in our two products being integrated in a timely fashion. We need to determine the resources of each of our respective development staffs to make such events occur. I feel confident that we can do this without draining resources from our current (difficult) development agenda. It will take some coordination and supervision, but not all the resources of a Greg or Kern or John. And I want to lead it.

[229] We continued to use Cadetron's original name for the product, "The Engineer Works", through the initial "Pioneer" marketing program. The product was renamed "AutoSolid" at the full production release, shipped in June of 1988.

Removing the Hardware Lock

With the introduction of AutoCAD release 2.1, all versions sold outside the United States and Canada were protected by the "hardware lock" or "WIDGET" (Walker's Inline Device Guaranteeing Elimination of Theft). This is a transparent RS-232 device which AutoCAD probes and requires to be present in order to run. When the introduction of this device went reasonably smoothly, Autodesk U.S. introduced it in the domestic market in release 2.5 in June of 1985. Never in our wildest imagination could we have anticipated the reaction. Suddenly we were exposed to a blast of vilification, moralism, and hypocrisy that (in my case, at least) forever changed the way I'll approach selling productivity tools to customers. All of the industry analysts and press people who had questioned us sharply about the threat of piracy in our market abandoned us to take the heat of trying to do something about it alone; not one word of support for us was written. Competitors jumped in to promote their products as better because they did not prevent theft, and products appeared on the market which claimed to defeat our lock, and were marketed "only to benefit the customer". The most notable of these products was itself copy protected.

On September 19, 1986 I recommended that we remove the hardware lock from the domestic product. After extensive discussion and preparation, we announced that the lock was being removed on November 25, 1986. The following document was distributed to all Autodesk employees on November 25; other than talking about the issue as a firm decision instead of as a recommendation, it is identical to the original memo I wrote to management recommending that we pull the lock.

We have continued to use the lock on international versions, and have encountered none of the problems we had in the U.S.

Why We're Removing the Hardware Lock

by John Walker
November 25, 1986

I think the time has come to admit that we made a misjudgement with respect to the hardware lock and plan an orderly retreat from the mess we have gotten ourselves into.

This paper will try avoid issues of morality and justice and focus on the business issues involved. I think we have all hashed over and debated the morality of this to exhaustion and in fact that's one of the reasons I make this recommendation. I will only reaffirm that I continue to believe that our software licensing policy of one user, one license is the only sound foundation on which a viable software industry can be founded; that the

fundamental problem with the hardware lock is that it prevents theft of our intellectual property; and that any moral opprobrium we direct at those who steal the product of our labours is fully shared by the manufacturers of computers who profit by selling machines which provide no form of protection for the property which makes them useful.

I think that we underestimated the hypocrisy, moralism, and disdain for intellectual property that exists in the United States. It is not enough that we provide ever-increasing functionality at incremental update prices tiny by comparison with any other industry; any attempt to fund ongoing development through incremental sales is seen as a "large, rich company" oppressing its small, struggling customers, the overwhelming percentage of whom signed or implicitly assented to a license agreement which our hardware lock only acts to enforce. We must not only tolerate looting, we must not attempt to prevent it. Ayn Rand called it "the sanction of the victim".

But we must recognise that we are only a software company, not a major force for morality in the world. We must make the decisions which will make our company and our products prosper and try to act in the right within the constraints of the real world. And it is on that basis that I base my argument here. I think that there is nothing we can do in the short term or medium turn to reverse the moral climate which opposes us presently. Only a long term shift in perceptions, aided by a concerted, united effort by all software providers and supported by hardware manufacturers (a signal example of which would be IBM, Compaq, and Apple pledging that all new PCs made after 1987 would contain a serial number chip) can help. A climate where falling hardware prices is presumed to cause lower software prices is one in which much education remains to be done.

The issue of intellectual property protection only seems to respond to this type of coordinated fix. It was not a revulsion with cassettes, a fee per tape or tape deck, or the FBI raiding pirate pressing plants that the music industry finally settled on as the solution to its problems: it took a new, uncopyable medium, controlled by a strict licensing mechanism and a high capital start-up cost, and a pledge by the hardware manufacturers to forgo revenue from making a medium permitting direct copies (the 44 Khz DAT agreement).[230]

In implementing the hardware lock with release 2.5, I think we not only misread the moral climate in the U.S., but we also made two public relations errors. First, we failed to get out front and promote the hardware lock as license enforcement without copy protection. In retrospect we should have hit the ground selling; explaining how the lock was central to the vitality of a local dealer and support channel, a part of our ongoing commitment to R&D and low cost updating of installed customers, and the alternative to the large-corporations-only site license mentality so many other software companies are adopting which ends up discriminating against the little guy who built this business.

Second, we failed to sell 2.5 as the massive update it was. It was always our intent to "spring" the hardware lock on a release so compelling that all users would be forced to upgrade. 2.5 is such a release, but we have not sold our customers on that fact. The demand for the DXF downgrade program is the most evident symptom of this fact. And of course our underpromotion of 2.5 was compounded by the focus on the hardware lock in most of the coverage of it.

I believe that it is too late to remedy either of these errors now. One, the discussion has become so polarised over the hardware lock, 2.5 is considered "the copy protected AutoCAD", and we would have to overcome a very high barrier of resentment just to be heard. Two, our key users already have 2.5, the reviews are in, and they are well into the phase of picking it apart, preparing wish and complaint lists, and looking to what Autodesk will do next. A major push to promote 2.5 as a new release would look odd and defensive at this date (I distinguish general promotion of AutoCAD as it stands, the central theme of all of our promotion which

[230] Which, in late 1987, seems to be unraveling under the very same kind of pressure we encountered.

should continue and be expanded, from specific "new and improved" promotion aimed at the installed base and at industry decision makers).

I think the central issues here are bad faith and good will. The small business users on whom our success has been based are guilty of what can only be called stunning hypocrisy and bad faith when they install additional computers at five to fifteen thousand dollars each but claim that theft of additional AutoCADs is the margin that keeps them out of bankruptcy. However, this market is the heart and the soul of our business and we should decide, and soon, if we want to debate with it or sell to it. Autodesk has over the last four years, accumulated a large reservoir of good will, respect, and trust among the small user community. Our dealer channel is successful largely because of this market. To the small user far more than any other, AutoCAD is CAD, and Autodesk is seen as the company on his side, as opposed to an IBM, Lockheed, or McDonnell–Douglas. We seem to be spending this good will at an extraordinary rate, and purchasing very little with it.

And that's the final argument: we've been accused to forsaking our ideals and focusing on "the bottom line". All right, let's do some of that. We have just completed the best quarter in the company's history, but we have always had a large bulge in sales after a new release. I would be very hard pressed to argue that hardware-lock-induced additional sales have contributed much to this sales performance, versus the normal post-new-release bulge we've experienced. There is a convincing counter-argument to this: that due to the hardware lock, our installed base has purchased only their first low-cost 2.5 upgrade. When, over the next 6 months they actually experience how useful 2.5 is, they will find the money to legitimise their additional copies and we will reap those sales. Were I an AutoCAD user today, I might well decide to wait and see if Autodesk crumbled under the pressure and removed the lock rather than ponying up $2750 each for my bootleg copies.

Having said all of that, my recommendation is based on these simple facts: we have not experienced a large increase in sales based on shipping the hardware lock; we are expending at a rapid rate and with little obvious return the good will of those most satisfied with our products and most influential in recommending additional purchases; we have failed to find strong support for the hardware lock among the very dealer community which is most benefited by it; our precious management, technical, and product resources have been diverted into a largely defensive effort; we are imperiling our perception in the market sector we most control at the very time that our lack of obvious technological leadership and growing competition from larger vendors puts our future most in doubt in those corporate and government accounts least likely to be worried about the hardware lock.

We should remove the lock now, on an incremental 2.5 update, because if we wait until the shipment of 2.6, all the publicity attendant upon that release will be buried under the news of our removing the lock. We lost almost all of the publicity on 2.5 enhancements in the debate over the lock, and we simply cannot afford to have the news of 2.6 buried in news of our reversal. We should concurrently go on the offensive with a promotion campaign explaining what we have done and why. This will act to palliate the inevitable blast of "Autodesk repents major marketing blunder" publicity which will attend our announcement.

I expect that the two weeks after our removing the lock will be very difficult weeks. I expect those who said that they would re-embrace us as the market leader if we removed the lock will remain silent, while those moralistic mountebanks who have been reaping profits larger than ours by far as a percentage of sales by selling products purporting to "break the lock" will crow over their "victory". Further, I expect some of the very dealers who have been silent or petulant about the lock will now view its removal as an assault by Autodesk on the viability of their businesses. And we will be assailed by publicity and cheap shots about our "blunder", "indecision" and the "shakeups in Autodesk". One of the principles I've always followed in business is that there's nothing wrong with being wrong—if you never try something that entails risk you're doomed to stagnation and eventual failure. Catastrophe is engendered by *staying wrong* in the face of clear evidence that you're on the wrong course. I think that we're far better off putting this episode behind us now. I believe that we are doing the

right thing in getting this over with and getting back to what we do best: developing, selling, and supporting products which revolutionise the way designers do their work.

Urgent Fury

By the end of 1986, AutoCAD's user interface was looking pretty antiquated. In October 1986 we'd shipped AutoSketch which had, for the time, a rather nice interface including pull-down menus and dialogue boxes. Many people felt that outfitting AutoCAD with such ease-of-use features would be a massive job requiring major modifications all over the program, further complicated by the need to maintain compatibility with existing menus, scripts, and other application components.

I wasn't sure it was all that tough. The moment the company shut down for the Annual Week of Rest between Christmas and New Year, I launched into a fury of round-the-clock programming, integrating the user interface manager from AutoSketch into AutoCAD, then extending AutoCAD's menu system to permit user-customisable pull-down menus and adding dialogue boxes to control many aspects of the program. By the time the company re-opened on January 4, 1987 it was all working as described in the following document which I stuffed into all the programmers' boxes so they'd find it when they got into the office that Monday. This was the first high-profile "Holiday Hack" and, along with the Portable Data Base project (see page 372), which I did the next month, formed the centrepiece of AutoCAD Release 9.

Urgent Fury

A new user interface can be added to AutoCAD in incremental steps without sacrificing open architecture. Installation of this capability may spark the next large growth in sales.

by John Walker
January 1st, 1987

THE EVOLUTION OF AutoCAD's user interface has, to a large extent, recapitulated the evolution of user interfaces within the computer industry since the advent of timesharing.

The first programs designed for timeshared computers were command driven. These programs required the user to learn and become facile with a fairly large set of commands expressed as words or abbreviations and entered on a keyboard. As time passed, programs came to provide assistance to the user in the form of on-line help, command completion, and user-definable macros.

As on-line systems were adopted for commercial applications in the 1970's, menu driven user interfaces became increasingly popular. In a menu driven system, the user's major act of volition is choosing from a set of alternatives presented by the computer. Such systems can lead the user through a maze of functions with minimal confusion. In such systems, however, it is clear that the computer is in charge and the user is at best a guide and at worst a peripheral.

Research on how untrained users, particularly children, perceive computers led to the development in the early 1970's of the first event driven user interface: the Smalltalk system developed by Alan Kay at Xerox PARC. Event driven systems superficially resemble menu based systems, but differ in that the user is in control, generating requests of the system which are performed regardless of its state. Event driven systems tend to use a flat command space model and are largely devoid of modes.

AutoCAD began as a completely command driven program, and was consequently an exemplar of the first class of timesharing applications. Immediately prior to COMDEX 1982, a screen menu was added to AutoCAD, complementing the preexisting rudimentary tablet menu facility. Commands generated by menu selections were allowed to pause for user input, allowing the development of simple menu macros. In 1983 and 1984 the AutoCAD menu facility underwent further development, culminating in the release of AutoCAD 2.0 in October of 1984. This package incorporated a true menu programming language, allowing replacement of menus and submenus, and supported four separate tablet menu areas, a button menu, and the screen menu simultaneously. This package, along with a hierarchically structured menu released in conjunction with it, advanced AutoCAD into the second generation of user interface.

Unlike many other programs which made this jump, AutoCAD added menu-driven capability on top of the existing command-driven architecture, preserving the preexisting interface for users who preferred it. Since AutoCAD is an open architecture program and has always encouraged its users to extend it, this was particularly important. An open architecture program such as AutoCAD is founded on the concept of its command language being a programming language. Only a command-driven language is well suited to interpretation as a programming language and the retention of AutoCAD's original command language, and its extension, first through menu macros and later with AutoLISP, contributed to the acceptance of AutoCAD as the standard for desktop computer aided design.

Many observers of the development of software believe that programs cannot evolve from one generation of user interface to another without being rewritten. Indeed, many programs have been rewritten concurrently with being fitted with a contemporary interface. In addition, the goal of open architecture as exemplified by an extensible command set, programmability, and mutability of the interaction with the user, is often viewed as detrimental to the goal of a consistent event driven user interface. Most event driven programs (the majority of which haven been developed for the Smalltalk system and its derivatives such as the Xerox Star and the Apple Lisa and Macintosh) have been closed architecture systems, providing little or no ability for the user to tailor the system.

Those who have extensively used command-, menu-, and event-driven systems commonly remark that these systems form a continuum. The command based system is the hardest for the new user to learn, but the most productive in the hands of the experienced "power user". The event driven system, though easily mastered by the beginner, delivers little more productivity to the user who uses it constantly. An open architecture package must also consider the consequences of the adoption of a new user interface on its extensibility and the base of applications which have been built upon it. Since the history of computing has demonstrated that open architecture, extensible systems predictably supplant closed architecture, proprietary systems, any modification of the user interface of a successful, established package must be carried out in an upward compatible, responsible fashion.

This article describes a set of modifications to AutoCAD which, taken as a whole, permit AutoCAD to present the user with a third generation event driven user interface. These modifications, while numerous, are individually minor. They build on AutoCAD's open architecture, making it the first true open architecture event driven program. The individual features described below work together in a tightly coupled fashion to deliver a far more responsive, intuitive, and transparent model of interaction with the user. They cannot be adequately evaluated by reading feature descriptions or by trying them in isolation. I recommend you see a demonstration of these new capabilities before attempting to understand the details of their implementation.

Some of these new capabilities require functionality not previously provided by AutoCAD display drivers, in particular the ability to save, restore, and clear rectangles on the screen, and to write text on the screen in any character cell position with normal, reverse, or disabled attributes. The display driver interface was extended to provide these additional functions in such a way that if an existing driver is not or cannot be upgraded, the new features will automatically be disabled, leaving existing AutoCAD functions unchanged. Several of the new features do not depend on the display driver extensions, and are present in all AutoCAD configurations.

The Menu Bar

A menu bar and pull-down menus are provided via extensions to the existing menu facility. Ten new menu sections are defined, named **POP1** through **POP10**. If the display driver is capable of supporting the new user interface, and the status line is configured, AutoCAD scans the **POP** menu areas and assembles a *menu bar* consisting of the first lines of each of the pop-up menu sections. If no ***POP*n* sections are defined, or the display driver lacks the extended functions, or the status line is deconfigured, the menu bar and pop-up menu facility will be automatically disabled. Any **POP***n* menu sections present will be ignored.

The menu bar is displayed by moving the pointer above the top of the graphics screen (exactly as the standard screen menu is activated by moving to the right of the graphics area). When the pointer moves off the top of the screen, the status line and first line of the standard screen menu are replaced by the menu bar. If the cursor is lowered back into the graphics area, the status line reappears. The menu bar may be accessed only with a physical pointing device—the arrow keys may not be used to display the menu bar.

When the cursor is moved over an item in the menu bar, it highlights by flipping to reverse video. If the pick button is pressed while a menu bar title is highlighted, a menu will appear on the screen below the title. Moving the cursor within a menu highlights items within it, and picking a highlighted menu item executes the commands associated with it. A pulled down menu remains down until either an item from it is picked, another menu is pulled down by picking it from the menu bar, the menu is removed by picking an unused section of the menu bar, or a point is selected by picking a point on the graphics screen, or a character is typed on the keyboard.

Pull down menus behave exactly as any other menu sections, and menu macros may be defined exactly as one would for the standard screen or tablet menus. Menu labels, which are limited to 8 characters for most display devices, may be of any length within **POP** menu sections.

Menu and submenu swapping via "**$**" commands may be used with pop-up menus. The pop-up menu areas are named **P1** through **P10** for menu swapping purposes. The following illustrates a menu swapping sequence.

```
***POP5
**P5A
[Submarine]
```

```
Ready
Aim
[Fire]$P5=P5B
Ready
[Steady]*^Credo
Go
[--]
[Test Sonar]^Cgrid on
[Raise periscope]^Csnap on
[Lower periscope]^Csnap off
[--]
[Crash dive]^Cquit y
**P5B 4
[Fire 2]$P5=P5C
**P5C
UpScope
[Torpedos]$P5=P5D
Running
    .
    .
    .
[Ping]$P5=P5A
**P5D 2
[~Empty]$P5=P5E
**P5E 2
[Torpedos]$P5=P5D
```

Since the title which appears in the menu bar is simply the first line of the menu, the title may be changed by replacing that line of the menu with a **$P***n***=** command. If the first line of the menu section is blank, no entry for it will appear in the menu bar. This allows you to turn whole menus on and off. This may not be a good idea in user interface design, but if you want to do it, you can.

If a menu item begins with a tilde ("~"), it will be displayed "greyed out", indicating that it is not a valid selection. The action taken when such a menu item is selected is up to the creator of the menu. If the item is a label with no commands, selecting it will have no effect. However, if you wish to have a greyed out menu item execute commands, you need only supply them. A menu item with a label of two minus signs is expanded to a separator line which fills the entire width of the menu. These items usually have no effect when picked, but again, if you supply commands to execute, AutoCAD will comply.

Extended Object Selection

To allow more convenient object selection, three new commands were added to AutoCAD's "Select objects:" mechanism. These commands may be used to adapt object selection to the form best suited to a given command. The design for these new facilities draws heavily on the object selection rules in AutoSketch.

Box selection. If you type **BOX** to the "Select objects:" prompt, AutoCAD will ask you to specify a window by two corners, as for the **WINDOW** and **CROSSING** commands. If the first point entered is to the left of the second point, **BOX** is equivalent to **WINDOW**; if the second point is to the left of the first, **BOX** is equivalent to **CROSSING**. Using **BOX** in a menu allows the user to choose between **WINDOW** and **CROSSING** modes simply

by how the box is drawn; no keyboard entry or further menu selection is required. The **BOX** command must be spelled out in full; no abbreviation is defined for it.

Automatic selection. Entering **AUTO** (abbreviation **AU**) to the "Select objects:" prompt chooses automatic selection. A point is requested as for single object selection. If the specified point chooses an object, that single object is the result of the automatic selection. If no object is chosen by this pick, it will form the first corner of a **BOX** selection as described above. Using **AUTO** in a menu permits the user to choose by pointing, window, or crossing simply by the way the pick is made.

Single selection mode. The specification **SINGLE** (abbreviation **SI**) places object selection in *single selection mode*. This disables the normal dialogue conducted by object selection, and causes it instead to return the first object or set of objects successfully selected by a subsequent command. The "Select objects:" prompt will continue to be issued until a selection is made, but that selection (whether a single object or multiple objects chosen by a window) will be reported without pausing for further interaction. For example, consider the following menu item:

```
[Erase]^Cerase single auto
```

When picked, this item terminates the current command and activates the **ERASE** command. The selection of objects to erase is done in **SINGLE** mode with the **AUTO** selection command. In operation, the user picks this command and either points to the single object to be erased, or points to a blank area and pulls a window (crossing to the left, enclosing to the right), around the objects to be erased. This is identical to the AutoSketch **ERASE** command, except that it does not repeat. **SINGLE** selection mode leads to more dynamic interaction with the user. A complex selection may be useful at infrequent intervals, but the selection dialogue was primarily a means of confirming the object of editing commands before they performed frequently irrevocable actions. Now that a general **UNDO** facility is available, this is much less important.

Menu Item Repetition

Once the user has selected a command, he is likely to use it several times before moving on to another. This is an outgrowth of how people use tools; you pick up a tool, do several things with it, then pick up another tool and so on. "If all you have is a hammer, everything looks like a nail." AutoCAD's normal interaction model forces the user to pick up the hammer before driving each nail. The automatic command repetition triggered by null input was implemented as a response to this problem, but it does not permit command options to be specified. This can lead to the undesirable situation (oft complained about) where the user chooses one of the variants of arc input from the screen menu and draws one arc, then repeats the command by hitting the space bar and is surprised to be back in the "three points" form of the **ARC** command.

Many systems are designed so that most of the frequently used commands are *modal*. That is, they repeat automatically until another command is chosen. AutoCAD was not designed on this model, but with the provision of transparent commands for most viewing and mode setting functions, it may be easily simulated.

An extension to the menu macro language has been made as follows. If the first character of a menu command string is an asterisk, and that command string is picked as the response to the "Command:" prompt, the menu string is saved. Subsequent "Command:" prompts will be answered by that menu string until it is terminated by the entry of Control C from either the keyboard or from another menu item. It can be seen that if all major commands are chosen from menus, begin with Control C, and use the repetition option, modal operation is achieved. The following essentially emulates part of the AutoSketch Edit menu.

```
***POP2
[Edit]
[Move]*^Cmove single auto
[Copy]*^Ccopy single auto
[Erase]*^Cerase single auto
[Stretch]*^Cstretch single c
[Rotate]*^Crotate single auto
[Scale]*^Cscale single auto
```

Multiple Command Modifier

While menu item repetition provides repeated execution of commands chosen from menus, the experienced user cannot repeat commands entered from the keyboard other than by improvising an AutoLISP macro on the fly. To provide this capability, a new **MULTIPLE** command is provided. If the **MULTIPLE** command is entered at the main "Command:" prompt, the next command entered is saved. When the "Command:" prompt reappears, that command will be automatically repeated unless a Control C was entered since the multiple command was activated. If a user wants to draw a bunch of circles, he may just say **MULTIPLE CIRCLE** and AutoCAD will repeat the **CIRCLE** command until a Control C stops it.

The **MULTIPLE** command does not issue a prompt, so the user is encouraged it view it as an adjective that modifies the next command. A console break will stop the iteration of a repeating command, so if you use **MULTIPLE** with a command that does not accept input, you can still stop it.

When a command is repeated by the **MULTIPLE** mechanism, the "repeat flag" which is set by repetition caused by a null input is *not* set. This was thought to be the most intuitive choice, as so few commands currently behave differently the user may not be aware of them and confuse their action with a bug in the **MULTIPLE** mechanism. Since the **MULTIPLE** command repeats only the command itself, parameters must be respecified on each subsequent execution.

Command Redefinition

Ever since the development of external commands via the **.PGP** file and user-defined commands in AutoLISP, users have repeatedly asked to be able to redefine built-in AutoCAD commands. Our concern in allowing this was that users would inadvertently render menus or AutoLISP programs unusable by redefining a command used somewhere in their guts. The facility implemented permits redefinition while guarding against the horrors it might lead to.

To delete the built-in AutoCAD definition of a command, use the **UNDEFINE** command. The command you name must be a known AutoCAD command, but may have been previously **UNDEFINE**d. To restore the built-in definition of a command, use the **REDEFINE** command, specifying the command to be restored.

Even if a command has been **UNDEFINE**d, it can always be activated by specifying its "true name", which is simply the command name prefixed by a period. The following sample dialogue illustrates these commands:

```
Command: (defun C:LINE ()
1> (princ
```

```
2> "Shouldn't you be using Polyline?\n")
1> (command ".line"))
C:LINE
Command: line
From point:  *Cancel*
Command: undefine
Command name: line
Command: line
Shouldn't you be using Polyline?
.line From point: nil
From point:  *Cancel*
Command: redefine
Command name: line
Command: line
From point:
```

Menus, scripts, and AutoLISP programs which are expected to be run in environments in which redefinition is used should protect themselves by using the "." forms of all commands. The **UNDEFINE** command operates only on main AutoCAD commands: those chosen from the "Command:" prompt. Subcommands, such as those used in the **LAYER** and **PEDIT** commands, and the dimensioning subcommands may not be **UNDEFINE**d.

Dialogue-Oriented Commands

A dialogue handler was installed in AutoCAD and used to implement several new commands which set AutoCAD modes via dialogue boxes. The dialogue handler is an extension of the one used in AutoSketch and AutoShade, and will be familiar in both appearance and operation to those who have encountered it before.

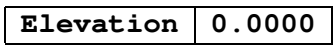
Since the dialogue handler requires the new display driver services, the commands which use it are only available on systems with a display driver which provides the new functions. Commands which use dialogues are flagged in the command table with a special prefix. Attempting to use one on a non-capable display will result in the message "**** Command not allowed with this display configuration. ****" and the command will be ignored. It is important that any new facility added to AutoCAD via a dialogue also be provided in the command language (or its equivalent, such as a system variable settable with **SETVAR**). This is essential for two reasons. First, users who lack a display which can show dialogues would be locked out from the new feature. Second, dialogues are only for direct on-line interaction with a user: AutoCAD facilities must also be made available from menu macros and AutoLISP, both of which must go through the command language interface. These constraints should in no way cause us to hesitate in freely using dialogues in AutoCAD: by presenting information to the user in a two dimensional form and by showing directly the interaction between features, they can improve the ease of use of the complex facilities that abound in AutoCAD by at least an order of magnitude.

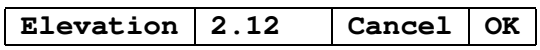
Dialogue-oriented commands simply pop up the dialogue, talk to the user, and exit. They should not perform any conventional (text scrolling line) user input or output. Since dialogue commands will typically be activated from menu items, they may be given obscure names unlikely to conflict with anything. I've adopted the convention that all dialogue command names begin with **DD** (for *Dynamic Dialogue*). Dialogue commands should be transparent wherever possible. All dialogue commands are presently ADE-3.

When a dialogue appears, the cursor changes from whatever it was to an arrow. Every dialogue has an **OK** button and most have a **Cancel** button as well. Until the dialogue is dismissed by picking one of those two buttons, AutoCAD responds only to pointer motion, pointer picks, and the keyboard. The screen menu (including the menu bar), and all of the other menu areas will not respond, nor may the flip screen or other mode toggle keys be used. The arrow keys may be used to make dialogue box selections, but it's hardly practical to work that way.

The act of popping up a dialogue, making selections, and placing them into effect by picking the **OK** button is one command from the standpoint of the **UNDO** mechanism. As with a transparent **SETVAR**, mode changes made within a command will be **UNDO**ne if the enclosing command is reversed.

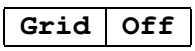
Most dialogues contain one or more *buttons*. Each button controls a value. Picking the button affects the value in a way that depends on the type of the button.



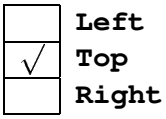


An *input button* specifies a value such as the snap spacing or the current drawing colour number. Picking an input button opens it and allows you to type values into it. Whilst open, an input button shows its own **OK** and **Cancel** subbuttons which may be used to accept or discard the value typed into the button. If you position the cursor over an input button so that it is highlighted, you may start typing into it without picking it. If an invalid value is entered, the **OK** button will have no effect; you should backspace, correct the value, and try again. Pressing the **RETURN** key is equivalent to picking the **OK** button, and pressing **ESC** is equivalent to picking the **Cancel** button. Values in input buttons are displayed and entered according to the settings of the **UNITS** command.

A *requestor button* looks like an input button, but when it is picked, the value area expands into an entire dialogue which is used to select the value for the button. When that dialogue is completed, the chosen value appears in the value area of the requestor button.



A *toggle button* controls a Boolean mode, such as whether the grid is shown on the screen. The value in a toggle button is shown as **On** or **Off**, and picking it flips it to the other value.



A *check button* is usually used to control selections from a set of alternatives, such as which isometric plane is to be used, or which layers are visible. A check button is a small rectangle which is either blank or shows a check mark ("√"). If a family of check buttons control mutually exclusive modes, such as the isometric plane, checking one turns the previously checked button off. If the controlled modes are independent, such as layer visibility, the buttons will be independent of each other.

Cancel

An *action button* does not control a value, but rather causes something to happen when picked. The **OK** and **Cancel** buttons are the most commonly encountered action buttons. Other action buttons perform such functions as scrolling a list of layer names up and down.

Errors within dialogues may result in the appearance of an *alert*. An alert is a small dialogue containing only a message and an **OK** button. After you read the message, get rid of the alert by picking the **OK** button.

DDRMODES command. The **DDRMODES** command pops up a dialogue which controls on-screen drawing aids. It subsumes all of the functions of the **SNAP**, **GRID**, **AXIS**, **ORTHO**, and **ISOPLANE** commands, and in addition provides direct control over **BLIPMODE**. The command is transparent.

DDEMODES command. The **DDEMODES** command displays a dialogue that sets the modes assigned to new entities. The functions of the **COLOR**, **LINETYPE SET**, **LAYER SET**, and **ELEV** commands are provided by this dialogue. The elevation and thickness are set by input buttons, and the colour, line type, and layer buttons activate dialogues used to choose the values from those known to AutoCAD. The **DDEMODES** command is transparent, but the values it sets may not take effect until the start of the next AutoCAD command. The only line types shown by the **DDEMODES** dialogue are those previously loaded in the drawing.

The Portable Data Base

The implementation of the portable data base in AutoCAD Release 9 finally completed the unification of the product across all machine architectures. The development notes describing this project are an example of the developer documentation that accompanied code submissions in the period.

The Portable Data Base

AutoCAD databases are now portable between operating systems and machine architectures. This allows efficient use of networks containing both personal computers and 32 bit workstations.

by John Walker
February 3rd, 1987

It was a dark and stormy night. The trees swayed in the wind, and the rain beat upon and streamed in rivulets down the dark window pane illuminated only by the cold light of a Luxo lamp, the flickering of a Sun 3 monitor, and the feeble green glow of a programmer debugging too long.

When the doorbell rang, I almost welcomed the interruption from the task in which I was engaged: fourteen subroutines deep in **DBXTOOL**, on the trail of a stack smasher which not only obliterated AutoCAD, but wiped the information the debugger needed to find where the error occurred. I glanced at the clock and noticed that it was 3:30. Since it was dark outside, it must be 3:30 in the morning. Only a very few people show up at the door at 3:30 on a Sunday morning.

Let's see: the stereo isn't on and no recent revelations have called for celebratory reports from the carbide cannon or the .45, so it's probably not the neighbors or the cops. That narrows the field considerably. I fully expected to open the door to see Kelvin Throop, as always slightly distracted, somewhat overweight, his face looking like it had been slept in, but sparkling with anarchic and subversive ideas.

With the usual irritation mingled with expectation, I opened the door and discovered I was looking at the neck of my early-morning caller. I looked up, and saw a face I had not seen for almost twenty years. It was a face free of pain and fear and guilt. John Galt had come to call in the middle of the night.

"Galt", I said, "I haven't seen you since, when was it, 1967? That's right, December 1967 it was. We were walking down the railroad tracks in Cleveland; the snow was a foot deep on the ground, the sky was grey and

the only warmth was the switchbox heaters at every set of points. Yes, it all comes back now. I remember you saying it was all over and you were going to drop out, and me saying things were just about to turn around. And I remember turning around and walking back to study for the physics exam and seeing you disappear into the snowy distance. Hey, come on in, have a Pepsi, tell me what you've been up to."

Galt walked in the door, put down his paper bag and, as always, strode to the refrigerator and opened the door. He poured a tall Pepsi and made a peanut butter, turkey, swiss cheese, and onion sandwich, polished both off, and then turned to me and spoke.

"As usual, you've got it all wrong. It wasn't December 1967, it was November—November 8. The first Saturn V launch was scheduled for the next morning, and you were bubbling over about how the final triumph of technology would turn around a disintegrating society. I said I'd had it with this decadent, exploitive culture, and I was no longer going to allow my mind to be enslaved by the looters. I tried to convince you to join me. But your time had not yet come. So I moved on to convince others, and to work on my speech."

"Hey, I remember that speech. How's it come since that draft I read back in '67."

"Pretty well. I'm up to 560 pages now, and there's no filler in there. I'm adding a refutation of the epistemology of Kant cast in terms of Maxwell's equations, and that will probably stretch it a tad."

"Don't you think that's a bit long?"

"Well, with the attention span of this society down to less than 30 seconds, some of the induction steps may get lost in the shuffle, but it's full of great sound bites and should play on the news for days."

"When 'ya gonna cut loose with it?"

"When the collapse of this decadent society due to its disdain for the products of the mind, and the consequent disappearance and exodus of the creators becomes self-evident."

"Hey, Galt, lighten up! When I last saw you the cities were in flames, the US was losing a hopeless war, the stock market had just crashed, the gold standard was being abandoned, three astronauts had died in a fire, the SST was facing cancellation, and the ABM was being negotiated away. Look at what you've walked out on! We have peace and prosperity, business is booming, and basic science and technology have flowered in directions unimaginable by the world in which we last spoke."

Galt walked into the computer room. He looked at the PC/AT linking AutoCAD. He looked at the Sun monitor, which was showing a full compilation of AutoCAD in one window, a completed execution of the regression test in another, and the debugger in a third. He walked over to my bookcase and pulled out my copy of the Dow Jones Averages chartbook from 1885 to the present. Moving in that eerie way he always did, in one motion he pulled the book from the shelf, opened it, and spread it in exactly the open space between the keyboards of the Sun and the IBM. For a full ten minutes Galt was silent as he turned the pages from 1968 through 1986. It appeared to me that the man had been out of circulation for a long time. I watched his face carefully to see if it registered surprise as he hit 1985 and 1986, but as ever those stony features remained unmoved. Galt closed the book, replaced it on the shelf, sat down on the chair in front of the AT, and turned to me. "Just wait," he said.

"So, enough about me", Galt continued, "what are you doing?"

"Well", I said, "where to begin? In '68 I. . . "

"Oh come off it," Galt interrupted, "I have my sources, after all. I mean what are you working on now."

Sheepishly, I continued.

Background

When we ported AutoCAD to non-MS-DOS systems, we were faced with numerous compatibility issues. Although all systems use the ASCII code, compatibility stops about there. Various systems have adopted different conventions for end of line and end of file detection; they store multiple byte binary values in different orders in memory, require different physical alignment of values on byte boundaries, and even use different floating point formats.

These issues make it very difficult for systems to interchange binary files. The only reasonable approach is to define a portable format, hopefully close to the middle point between the systems, then require every system to convert that format to and from its own computational requirements.

Our existing (2.5 and 2.6) AutoCAD releases do not allow interchanging binary files among major machine types (current major machine types are MS-DOS, Apollo, IBM RT PC, Sun, and Vax). To move data between systems, one must convert it to ASCII form, possibly translate the ASCII file due to end of line conventions, then load the file onto the other system and convert it back to binary. For drawing databases, this means one must **DXFOUT** on the sending system and **DXFIN** on the receiving system.

Given the difficulties in physically moving files between systems, the small market initially anticipated for non-MS-DOS AutoCADs, and the major work needed to make binary files portable, we chose not to address this problem previously. Sales to date of non-MS-DOS machines indicate that this decision was correct.

The advent of high speed networks and file sharing protocols such as Apollo's Domain, DEC's Decnet/Vaxmate, and Sun's NFS have begun to erode the justification for this decision. Many AutoCAD users, particularly in larger companies, have inquired about configurations involving a file server, one or more 32 bit workstations, and a number of MS-DOS machines, all on a common network. Such a configuration economically provides large central storage, high performance when needed, and very low cost individual workstations for routine work. The usefulness of such an installation is drastically reduced if every transfer of a drawing from a PC to a 32 bit workstation requires a **DXFOUT** and **DXFIN**, as these are lengthy operations which consume a large amount of disc space and network bandwidth. As we increase our sales efforts in large accounts, a competent solution to the issues raised by heterogeneous networks will be a major point of distinction which can distance us from the competition.

The first step toward a compatible database was taken when Bob Elman redesigned the entity database code in release 2.5. Galt broke in, "*The* Bob Elman". "Yes", I responded, and showed him the listing of **EREAD.C**. He shook his head and said, "That's Bob". Bob's code resolved all issues of byte ordering and alignment in the entity data portion of the database, and did it in a particularly efficient way that takes advantage of the properties of the host machine's architecture. Entities are written with no pad bytes and Intel byte ordering. Thus MS-DOS machines, the overwhelming segment of our market, pay no speed or space penalty. Bob's code did not address machines with non-IEEE floating point (the VAX is the only exemplar of this class).

Providing drawing database compatibility between machines, then, is primarily an issue of fixing the drawing header record (**MASTREC**), the symbol tables (**SMIO**), and the headers on the entities themselves, plus resolving the issue of differing floating point formats. In addition, the other binary files that AutoCAD uses, such as

DXB files and compiled font and shape definitions should be made compatible. The work described herein defines canonical forms for these files, implements a general package for system-independent binary I/O, and uses it to make AutoCAD drawing databases and the other aforementioned binary files interchangeable. The code has currently been installed and tested on MS-DOS and Sun systems, which may now share files in an NFS environment. The work needed to port it to the Apollo and RT PC should be minor. A VAX version will require certification of the code to interconvert VAX and IEEE floating point formats.

Galt interrupted, "So what you're saying is that before, if you hooked big ones and little ones together on a wire, it was a pain in the neck, and now you've fixed it so it isn't".

For a longwinded pedant, the man does have a talent for coming to the point.

The Binary I/O Package

To read and write portable binary files, include the file **BINIO.H** in your compilation. You must include **SYSTEM.H** before **BINIO.H**. **BINIO.H** declares numerous functions, which are used to read and write binary data items on various systems. Each of these functions is of the form:

b_{r|w}*type* **(***fp*, *pointer*[, *args...*]**);**

where *type* is the mnemonic for the internal type being written, *fp* is the file pointer, *pointer* is the pointer to the datum being read or written (must be an lvalue), and *args* are optional arguments required by some types. For example, when writing a character array an argument supplies its length.

Thus, to write a real (double precision floating point) number **val** to a file descriptor named **ofile**, use:

```
stat = b_wreal(ofile, &val);
```

Each of these routines returns the same status **FREAD** or **FWRITE** would: 1 for single item reads and writes, and the number of items transferred for array types. Currently defined type codes are as follows:

char Characters. Signed convention is undefined. Canonical form in the file is a single 8 bit byte.

uchar Unsigned characters. Used for utility 8 bit data. Canonical form in the file is a single 8 bit byte.

short Signed 16 bit integers. Canonical form in the file is two's complement, least significant byte first, most significant byte last, two total bytes.

long Signed 32 bit integers. Canonical form in the file is 4 bytes, starting with the least significant byte and ending with the most significant byte. Two's complement.

real Double precision floating point numbers. 8 bytes in a file. Canonical form in the file is an 8 byte IEEE double precision number, stored with the least significant byte first and the most significant byte last.

string An array of **char** items. The third argument specifies the number of characters to be read or written. Canonical form in the file is one byte per item, written in ascending order as they would be addressed by a subscript.

If the binary I/O package is to do its job, you must be honest with it: only pass the functions pointers of exactly the type they are intended to process. If you use **b_wstring** to write a structure, you're going to generate

files just as incompatible as if you used **fwrite**. And you must never, never use an **INT** as an argument to one of these routines.

When using the binary I/O package, you must explicitly read and write every datum: there is no way to read composite data types with one I/O. Bob Elman's code in **EREAD** solves this problem by packing data into a buffer, then writing it with one call. Since this handles the entity data, which is by far the largest volume of data that AutoCAD reads and writes, I felt that taking a simpler approach in the binary I/O package would have no measurable impact on performance. I felt that the complexity of the mechanism in **EREAD** was not required for handling the other files.

On a system such as MS-DOS, whose native internal data representation agrees with the canonical format of the database file, the various read and write functions are simply **#define**s to the equivalent calls on **FREAD** or **FWRITE**. The variable **TRANSFIO** in **SYSTEM.H** controls this. If it is not defined, all of the binary I/O routines generate in-line calls on **FREAD** and **FWRITE**. If **TRANSFIO** is defined, machine specific definitions in **BINIO.H** are used to define the I/O routines. Compatible types such as **char** may still generate direct I/O calls, but incompatible types should be defined as external **int**-returning functions.

If a machine uses a non-IEEE floating point format, the **b_rreal** and **b_wreal** functions must convert the IEEE format in the file to and from the machine's internal representation. In addition, because the entity data I/O code in **EREAD.C** does not use the Binary I/O package, you must tell it to perform the conversion. You do this by adding the statement:

```
#define REALTRAN
```

in the **SYSTEM.H** entry for the machine. This will generate code within **EREAD.C** which calls two new functions your binary I/O driver must supply. Whenever a real number is being written to a file, **EREAD** will call:

realenc(*bufptr*, *rvalue***);**

where *bufptr* is a "**char ***" pointing to an 8 byte buffer in which the canonical IEEE value should be stored (remember, lsb first), and *rvalue* is the real number value to be stored, passed in the machine's internal type for **double**. When a number is being read, a call is made to:

rvalue **= realdec(***bufptr***);**

in which *bufptr* points to an 8 byte area containing the IEEE number. **Realdec** must return the corresponding internal value as a **double**.

Each machine architecture must define a binary I/O driver providing the non-defaulted I/O routines, and if real number conversion is required, **realenc** and **realdec**. Examine the driver for the Motorola 68000 family (**BIO68K.C**) for an example of such a driver.

Modifying AutoCAD

Utilising the binary I/O package within AutoCAD to implement portable databases involved modifications in several areas. The changes are large, numerous, widespread, and significant, despite their limited impact on what gets written into the file. Installing them and debugging database compatibility was not a difficult design task; it was simply a matter of hacking, slashing, slogging, and bashing until every place where a nonportable

assumption was made was found, and then fixing them all. "That's what you were always best at," Galt interjected. I said that I hoped so, for I know of no single project I've done within AutoCAD which is so likely to destabilise the product as this one. The following paragraphs cover the highlights of each section.

The Drawing Database

Making drawing databases compatible consisted of several subprojects. The result of all of this is that an AutoCAD with the new code installed can read existing drawing databases written by the machine on which it is executing, old MS-DOS databases, and new portable databases. It writes new portable databases, which can be read by any AutoCAD with this code installed.

The ability to read both formats of databases is implemented via the flag **rstructs**. When a drawing database header is read by **MVALID**, if it is an old, nonportable database, **rstructs** is set to **TRUE** and the file pointer used to read the file is saved. Subsequent reads from that file will use the old code to read aggregate data. At the end of every database reading operation, such as **INSERT** or **PLOT**, **rstructs** is cleared.

The drawing header. The drawing header is managed by code in **MASTREC.C**. The header is defined, for I/O purposes, by a table called **MTAB**. This table previously contained pointers and lengths for all the items in the header, and each was written or read with an individual call on **FREAD** or **FWRITE**. Compatibility problems were created by the fact that the header contained several kinds of composite objects: symbol table descriptors, transformation matrices, the "header header", a view direction array, Julian dates, and calendar dates. I modified the table to contain an item type and implemented a switch to read and write each item with the correct calls on the Binary I/O package. Special code had to be added for each composite type to read and write it; just adding entries to the table for the components of the composite types falls afoul of the mechanism that allows addition of new fields to the header. I tried it; it doesn't work. The symbol table descriptors have a several unique problems: first, their definition contains a "**FILE ***" item. The length of this item varies depending on the system's pointer length, so the structure changes based on this length. On MS-DOS systems, data in the structure totals 37 bytes, and different compilers pad this structure differently. The file pointer field means nothing in a drawing database, but it is present in all existing databases and it varies in length. But if you think that it never uses a pointer read from a file, you haven't looked at the code in **WBLOCK.C** that saves and restores the header around its diddling with it. Look and see the horror I had to install to fix that one.

The symbol tables. The symbol tables, managed by **SMIO.C**, were an utter catastrophe from the standpoint of portability. The problems encountered in **MASTREC** with their headers was only a faint shadow of the beast lurking within **SMIO**. To refresh your memory, each symbol table has a descriptor which is usually in the drawing header (another symbol table is used for active font and shape files, but it is not saved with the drawing and does not enter this discussion). The descriptor for the symbol table contains its length, the number of items in the table, the file descriptor used to read and write it, and the address within the file where it starts. There is no type field in a symbol table. Symbol tables are read and written by the routines **GETSM** and **PUTSM**, which are passed the descriptor. Each symbol table entry consists of a structure containing several fields of various types.

Previously, **GETSM** and **PUTSM** did not care about the content of the symbol table record; they just read and wrote the structure as one monolithic block. That, of course, won't work if you want the tables to be portable: each field has to be handled separately with the Binary I/O package. So in order to do this, **GETSM** and **PUTSM**

must know the type of table they are processing.

"So," said Galt, "add a type field to the table."

"Heh, heh, heh," I said, walking over to the Sun and bringing up all the references to the block symbol table descriptor in **CSCOPE**. There are few data structures within any program that are chopped, diced, sliced, shuffled, and maimed as much as an AutoCAD symbol table descriptor. Most (but not all) live within the drawing header. They can point to their own file or be part of a monolithic database. They contain that ghastly variable length file pointer which gets written in the drawing header. They get copied, created dynamically in allocated buffers, and in **WBLOCK**, saved to a file, modified to refer to another file, then read back in. And that "length" field I mentioned, **sm_eln**. Well, it may include a trailing pad byte on the disc depending on which compiler and options made your MS-DOS database. And it gets used both to seek into the file and to dynamically allocate symbol table descriptors except in the places where it uses **sizeof(struct** *whatever***)** instead. One week into this project, I had the feeling that I had not stuck my head into the lion's mouth—I had climbed into the lion's stomach.

The most severe fundamental problem was that I had to both decouple the symbol table descriptor on disc from the one in memory, and also introduce separate lengths for the symbol table as stored on disc (used to seek to records) and in memory (used to allocate buffers, copy tables, and so on). I ended up adding two fields to the symbol table item in memory, **sm_typeid** and **sm_dlen**, which specify the type of the symbol table (mnemonics are defined in **SMIO.H**) and its length as stored on the disc. When a symbol table is in memory, **sm_eln** specifies the length of the structure in memory. When a symbol table is written out, the two new fields are not written: instead the disc length is written into the **sm_eln** field and the type is expressed implicitly in the symbol table's position in the drawing header.

By the way, the lack of a type code in symbol tables has been felt before: there is some marvelous to behold code in **WBLOCK.C** that figures out which symbol table it is working on by testing the pointer against the descriptor address. I did not fix these to use my new type codes. Somebody should some day. Once the type codes and disc lengths were present, the changes to process the symbol tables separately were straightforward to install in **SMIO.C**.

Because the code to process the symbol tables field by field is substantially larger and also somewhat slower than reading a single structure, I set up conditional compilation to use the old code on MS-DOS. Since MS-DOS already writes the tables in canonical form and has the most severe memory constraints, there's no reason it should have to pay the price of compatibility code it doesn't need. Note that if you remove the **#ifdef**s on **MSDOS** from the file, it will still work fine: it will just be bigger and slower.

The entity headers. There is a fixed set of fields which precedes every entity in the drawing database to specify its type, flags, length of the packed data which follows, and a pointer. When Bob made the entity data compatible, he could not use his scatter/gather mechanism for these fields because they *control* the scatter/gather process. I modified **EREAD.C** to use the Binary I/O package for these fields. In addition, if **REALTRAN** has been defined on this system, the **gathreal** and **scatreal** functions call **realenc** and **realdec** routines to translate floating point formats. If **REALTRAN** is not defined, no additional code is compiled or executed, so IEEE-compatible systems pay no price for the possibility of floating point format conversion. *The floating point conversion mechanism has never been tested.*

Shape and text font files

Compiled text font and shape files were made compatible by using the Binary I/O package within **SHCOMP.C** when compiling a shape file and in **SHLOAD.C** when loading it. The shape files written by the modified code are identical to those generated by an MS-DOS AutoCAD but are incompatible with other systems. All **.SHX** files on non-MS-DOS systems must be recompiled when converting to this release of AutoCAD. Attempting to load an old format file results in an I/O error message. It was my judgement that considering the tiny installed base of non-MS-DOS systems, it just wasn't worth putting in some form of level indicator and generating a special message. *This code has never been tested with "big fonts" (e.g., Kanji).*

DXB files

Binary drawing interchange files were just plain busted on non-MS-DOS systems. The problems were:

1. Type codes greater than 127 did not work due to some code incorrectly copied from **SLIDE.C**.
2. An **fread** was done into an **int**, resulting in failure on any machine whose **int**s are not 16 bits.
3. The AutoCAD manual documented **.DXB** files as being in Intel byte order, but the code did not perform the required reversals.

I modified all I/O within **DXBIN.C** to use the Binary I/O package, and corrected these problems. All systems now read **DXB** files which are compatible with existing MS-DOS files. Since the existing code in non-MS-DOS systems could never have worked, compatibility with existing non-MS-DOS **DXB** files is not a consideration since none exist.

Slide files

I corrected a problem in my earlier submission of code to make slide files portable which was found by the regression test. A null slide file created by MS-DOS (or the new portable code) would get an I/O error if you attempted to view it on a Sun. **SLIDE.C** was reading the in-memory length of the slide file header when it validated the header. I changed it to read the portable length in the file.

Compatibility status summary

The following is a summary of AutoCAD file portability as of the integration of this code.

Drawing files	Fixed to be compatible. All systems read both their own old-format files and the new portable files. All systems emit portable files.
ASCII files	Fixed to be compatible. Note that this causes the following file types to become compatible: **HLP**, **HDX**, **SHP**, **DXF**, **DXX**, **MNU**, **PAT**, **LIN**, **PGP**, **MSG**, **LSP**.
ACADVS	The virtual string file is compatible by design.
Filmrolls	Compatible by design.
Slides	Fixed to be compatible. Systems can read their own old files and portable files. All write portable files.
Slide libraries	Compatible by design.
SHX files	Fixed to be compatible. Old MS-DOS files are portable. Old non-MS-DOS files must be recompiled.
IGES files	Compatible by design.
DXB files	Fixed to be compatible. Previously worked only on MS-DOS. Old MS-DOS files work without modification.
MNX files	Incompatible. A system must use menus compiled by its own AutoCAD.

Upper and lower case

I have done nothing in this project to resolve the issue of case conventions for file names. I consider this issue so controversial and politically charged that I'm not yet ready to step into it. I hereby submit my recommendations for comment. Each system will define a tag in **SYSTEM.H** called **CASECONV**. It shall be set to one of four values:

CCMONOU System is monocase and uses upper.
CCMONOL System is monocase and uses lower.
CCULU System uses both cases and prefers upper.
CCULL System uses both cases and prefers lower.

When a system writes a drawing database, it stores its **CASECONV** setting in the drawing header. This is referred to as the "case convention of the sending system". When a system reads a drawing, if it was created on a system with a different case convention, it processes file names in symbol table entries based upon a matrix of the sending system's case convention and its own. If the receiving system is monocase, file names in symbol tables are not translated, but **FFOPEN** and its clones translate all file names to the receiving system's case convention

before submitting them to the system. If the receiving system uses both cases and the sending system was monocase, names in symbol tables are *translated at read-in time* to the preferred case of the receiving system. The names are then used as modified, without further modification by **FFOPEN**. This is asymmetrical and impossible to justify except by convincing yourself that this is the best approximation to what's best for the user.

My throat was feeling a little dry after such a lengthy dissertation. I got up to refill my glass. When I walked back to my chair, Galt was flipping through the listing of **SMIO.C** next to the Sun. He turned to me and said, "Why do you do this? Here you are in the middle of the night struggling trying to trick this megalith of software into threading its way around incompatibilities between computers that aren't even of your making."

I replied, "Differences in products are a consequence of their rapid evolution in a free market. Incompatibility is the price of progress". John Galt was speechless for at least 12 seconds.

He rose and said, "Join us. You weren't ready in 1967. Now, in 1987 you should see that you're struggling to make money in a world where the money you make is taxed away and handed to defence contractors like Lockheed and McDonnell-Douglas, who turn around and compete against you with products your taxes paid to develop. While so many others are sleeping, you labour to produce intellectual property, then you listen to others lecture you on their "right" to steal it. Can't you feel the circle closing? Can't you see that this can't go on? Why not hasten the inevitable and pave the way for a brighter day? You should drop out, or work to hasten the collapse."

I looked at the **DIFF**s of my portable database code. I said, "After this project, I can't help but feel that hastening the collapse would be an exercise in supererogation."

Galt shrugged. He sat back down and said, "Your time hasn't yet come. I try to talk to people when they'll see the issues most clearly. I try to find the times when they see what they're doing and begin to wonder why. I'll be back. It may be in two days, two years, or maybe twenty years."

We talked for an hour or so about old times, common friends, and shared interests. He left as the sun was rising.

Breaking the 3D Logjam

AutoCAD Version 2.6, which shipped in April of 1987, extended the pure extrusion "3D Level 1" of Release 2.1 by adding two new entities, 3DLINE and planar, three or four sided 3DFACE which could have any orientation in space. These were jammed in as a stopgap so that more interesting models could be created and rendered in conjunction with AutoShade. After Version 2.6, we were working in parallel on two projects. "White Album" (Release 9) centred around the pulldown menu and dialogue box features I had implemented in "Urgent Fury", my 1986–1987 "annual week of rest" project (this tradition is described on page 505), along with the Portable Database code (see page 372). The other project, called "2.7", was to be a fundamental redesign of the guts of AutoCAD to turn it into a general 3D modeler.

The 2.7 project, as originally constituted, was making very little progress, since attempting to re-architect the internals of a product as complicated as AutoCAD while preserving complete compatibility is an extremely difficult task, especially when it involves ripping up the product in such a manner that nothing works for a long period of time. By the time 2.6 shipped, the 2.7 project was going nowhere and everybody was getting discouraged.

In March 1987 Scott Heath suggested that we abandon the bottom-up rewrite approach and instead add 3D to AutoCAD in an incremental manner. This made a lot of sense—the design he was proposing seemed eminently doable, and we'd be able to work on the release just as we'd developed previous updates to AutoCAD.

I decided that breaking the 3D logjam would require not just a good technical design, but also punching through the discouragement that was beginning to persuade people "we just can't do 3D at Autodesk." So, indulging my flair for the dramatic, I decided to call a meeting, on a Sunday, at my house—in the same living room where Autodesk was founded—to figure out how to make AutoCAD do 3D, and how to get it done quickly. To reinforce the importance to the company, Al Green attended the meeting.

This memo was taken, by some, as "undermining the technical management," which kind of surprised me, since the technical management was getting nowhere in 3D and I, as chairman of the company, knew that if we didn't have 3D soon we'd be out of the game. Scott had made a concrete proposal for how to build a 3D AutoCAD, and my reaction was to let him try to pull it off. This was the genesis of AutoCAD Release 10.

Memorandum

To: Distribution
From: John Walker
Subject: Programmers' Meeting
Date: March 5, 1987

Autodesk has lost the technical leadership in the industry that it enjoyed for the four years in which AutoCAD became established as the worldwide industry leader for CAD.

The reasons for this loss are many. Seeking to assign blame or to revisit decisions made in the past with inadequate information squanders precious resources we should be spending to regain the high ground and continue our leadership of the industry.

We must announce a full 3D version of AutoCAD at COMDEX Fall 1987, and ship it before December 15, 1987.[231] If we fail to do this, we will forfeit our industry leadership, market share, stock valuation, dealer loyalty, and preeminent position as the application base for the CAD industry.[232]

I have invited you to this meeting because I believe that you are one of the cadre of people who can make this happen. Programmers created Autodesk because we believed that our efforts could create products and thereby create a company much better than the companies we worked for in the past. Now programmers must come to the aid of Autodesk in the time of its greatest technical challenge.

Scott Heath has made specific recommendations as to how we can implement 3D within AutoCAD in an expeditious fashion. Tony Monaco[233] has said that *nothing* is as important as delivering a full 3D system as soon as possible. The management and directors of Autodesk are unanimous in supporting Scott's initiative.

At 14:00 on Sunday, March 8, 1987, at my house (16 Saint Jude, Mill Valley), I will convene a meeting of those us who can help Scott to achieve this goal he has set for himself and our company. One thousand eight hundred and sixty three days before this meeting, sixteen programmers met in the same room and created Autodesk. This Sunday, another small group can meet there and rescue it.

I am not inviting you to enter into an easy task—it will be the greatest challenge we've undertaken and I expect it to be for most of us the most stressful and difficult task we've yet encountered.

I invite you to partake of a challenge that few people ever have a chance to undertake—a test of our individual and collective mettle which, if met, will yield not only material rewards commensurate to the efforts expended, but also that inner confidence that can come only from looking down into the black gulf of failure, then jumping it.

Please let me know as soon as possible if you're in. *No opprobrium will be attached to passing this up.* Parking is catastrophic, so I'll coordinate car pools up the mountain. Pepsi will be supplied; pizza will be ordered as

[231] This was not to be. Adding 3D to AutoCAD was a *big* job, and Release 10 did not actually ship until October 1988.

[232] There's Chicken Little Walker getting shrill again. I do think that if we'd slipped 3D much past 1988, we'd have been in real trouble.

[233] Vice President for Marketing and Sales at the time.

required.[234] Expect to leave this meeting (maybe 48 hours later) with specific task assignments which constitute the most challenging tasks of your career.[235] Expect to fly back from Las Vegas this November knowing that your efforts saved Autodesk.[236]

Distribution: Dan Drake, Bob Elman, Al Green, Ben Halpern, Patricia Harris, Scott Heath, Jack Heller, George Krucik, Duff Kurland, Mauri Laitinen, Greg Lutz, Eric Lyons, Tom Mahood, Steve Mighetto, Lars Moureau, Vladimir Pochop, Hal Royaltey, Kern Sibbald, Walt Spevak, Robert Wenig.

[234] When Eric Lyons called Domino's to order something like 16 large pizzas, they didn't believe him. He had to drive down the hill and order them in person.

[235] Actually, things were wrapped up by about 8 P.M.

[236] And by next November, they had.

Jeremiad

Every now and then I get my fill of arrogance, smugness, yuppies, blithering airheads, carpeted walls, opulent furniture, "departmental priorities", and "finalizing the game plan". Here's what happens shortly thereafter.

Remarks for the April Company Meeting

by John Walker — April 3rd, 1987

I was looking at some comparisons between companies a couple of days ago, and they set me to thinking.

Today, the market value of Autodesk, the number of shares of stock times the price per share, is about 610 million dollars. The market value of Computervision, for over a decade the colossus of the CAD industry, is now about 520 million dollars. In the minds of those who back their opinions with their money, Autodesk is worth more than Computervision.

How did that happen?

It happened because we were aggressive, hard-scrabble, hungry rats, responsive to our customers, and able to turn on a dime—while they were complacent, smug, arrogant, arthritic, and hubristic.

While we were working by day to tell our story, sign up dealers, help our users, convince OEMs, and manufacture quality products and ship them on time, and toiling by night designing, developing, testing, and innovating products, they were sitting around their fancy conference tables in their tasteful executive offices writing and reading their five year strategic plans, their market segmentation analyses, their technology forecasts, and their slick promotional brochures.

And while the dinosaurs congratulated each other on their success, we rats were eating their eggs. According to Charles Foundyller of Daratech, in 1982 every one of the top 10 CAD vendors was profitable, and they collectively forecast continued steady growth in unit sales, revenues, and profits. Today, five years later, more than four times as many CAD workstations have been sold as their forecasts predicted. Despite this, only 3 of the top 10 CAD vendors are profitable, and several are effectively leaving the business.

What happened?

Autodesk happened. In the space of 60 months, we plucked the technology of CAD from the clenched fists of the elitists and handed it to the tens of thousands of individuals they disdained. And the results are destroying the companies run by those complacent administrators in their serene office towers.

And what were those mandarins of CAD writing in their strategic overviews five years ago? What were they telling each other at their endless product planning meetings? Well, I'll bet they didn't even mention PC CAD or the low end of the market. I expect they were reassuring each other with those comforting phrases, "Now that we're a big company we can't react as fast as we did in the old days". "You can't expect people to work as hard now that we've made it as when we were struggling". "The low end of the market isn't viable—value will always command a premium price". "We have to protect our installed base—we can't afford the risk of new products to our sales". "The user can't adapt to new technology as fast as it is developed—we must deliver change to users at a moderate rate, one they can absorb".

We sold a hundred AutoCADs, then a thousand, then five thousand, and more. In 1984 AutoCAD became the most widely used CAD system in the world. And still they sat in their fancy chairs, skimming their executive summaries while dozing off to the drone of the projector fan. "The low end of the market may be high volume, but there's no revenue or profit there". "The serious end-user will demand more power.". "Our reputation and position in the market will always distinguish us from the little guys".

We twisted, we turned, we changed course, we took chances; we won and lost bets in the marketplace, we reacted to technological opportunities and user requests, and even though we told everybody exactly what we were doing, we managed to saw off the entire low end of the market pyramid before those statesmen of the CAD millennium got their minds off their Mercedeses, their stock bonus plans, their new office layouts, and their corporate image development programs and realised that we had rendered them impotent and obsolete.

Every quote I have put in the mouths of the big CAD people is a paraphrase of something I've heard in this building in the last two months.

I look at how our architects and interior decorators have betrayed, subverted, and sold out the low-rent surroundings which were not this company's image, but its very soul, and it makes me want to puke.

When we went to AutoCAD Expo in Europe, we went business class. I guess it's expected these days. The wide seats, the free booze, and the quiet surroundings cost our company a total of $27,000 more than the plebeian seats in the back of the plane. That's just a little less than half the cash it took to start this company.

Charles Foundyller showed another slide in his talk. It was a unit market share chart of the CAD industry, and Autodesk was in the usual number one position. Number two was Generic CADD, with half our market share. In the last year they have gone from zero to half as many units as us, and at the present shipping rates, we will end this year as number two.

Of all of the companies who sell products in the price category of AutoCAD, only Autodesk lacks a true 3D product. And we decline to even estimate a date for delivering one. "The customers have to understand, AutoCAD is a big, complicated program and... blah blah blah".

Why is this company successful?

Because more than 80,000 individuals and businesses have parted with thousands of dollars of their hard earned money to buy our products. They bought our products and the hardware to run them because they believed us when we told them that what we had to sell would do the job and repay their investment. Everything else flows from this simple act of trust and our keeping the promises we make when we sell a product.

If I hear one more comment about "end-user perceptions" I may be moved to homicide. What the hell is an end-user anyway? The last guy on a dull needle? I suggest that we ditch this yuppie-babble and call the people whose dollars and trust support our company by the old-fashioned term of respect they are due. Customers.

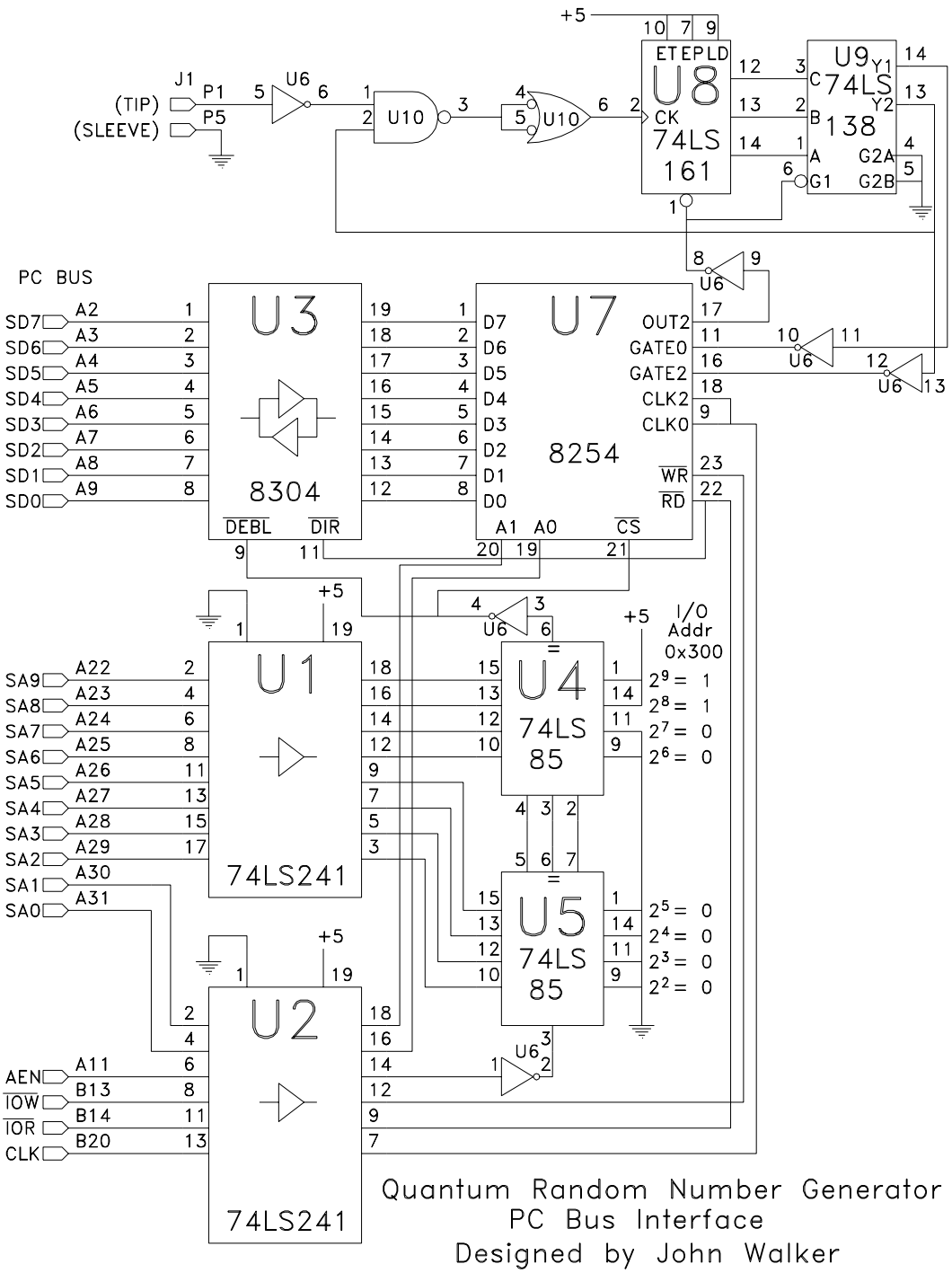
They deserve the very best we can do for them. Little else matters very much.

Now I'm sure that after sharing these thoughts with you—these thoughts that keep me awake night after night—some people will tell me that I've created an "atmosphere of crisis", and that they feel "insecure". Well they should. This is a high technology business, and the only thing we have to sell is the quality of our products and our service to the customer. Doze off, get smug, bet the wrong way, and it can turn on you overnight and wreck all of your well-crafted plans. Lose that ineffable edge that comes from dedication, concentration, and maximum effort, and the flaming knife of retribution will flense the extravagance, the triviality, and the arrogance from the carcass of a company on the way down. All of the Cross pens, all the conference tables, all the business class airfares, the volleyball leagues, and the carpeted walls in the world won't buy us a second chance if we forget and abandon the things that brought our great success. The risk today is no less and no more than it was when we started the company. It's been that way throughout the company's history. The big rewards come from the big risk, and if you can't accept that, you might be happier working for the phone company or the government.

I have here a relic of the founding of Autodesk—a sheet of our original stationery, which Roxie and I designed on our dining room table. Below the company name, it said "Excellence in Computer Software". That is what this company is about. Excellence in design, implementation, testing, documentation, training, marketing, sales, support, manufacturing, shipping, customer service, dealer support, finance, administration, and everything else we do. That is what got us here. That is what can keep us growing, changing, and continuing to create challenges, opportunities, and rewards for all of us involved in this venture.

This is about more than just making money. I think that what we are doing here is right. I believe that the rewards we have reaped so far stem from that rightness as much as from hard work and luck. Responsiveness to our customers, aggressiveness in developing our products and promoting them in the marketplace—excellence in everything we do. We owe our best efforts to continuing the development of this industry we have created in the last five years. We owe it to our customers. We owe it to our dealers. We owe it to ourselves.

Even though Autodesk taught me that selling software works much better than selling hardware, I've learned that success with software doesn't assuage the compulsion to build a gizmo every now and then. In 1986 I built this card that plugs into the IBM PC bus and attaches to a trivially-modified Heathkit RM–4 Geiger tube radiation monitor. I use a 10 μCurie ^{137}Cesium source to generate radioactive decay events. Along with a little software, this generates true random numbers, as opposed to computer-generated pseudorandom numbers. I drew the schematics using a development version of AutoCAD Release 9 in an attempt to find AutoCAD bugs. I didn't find a one.



Urgent Worry

After the March 8, 1987 programmers' meeting which launched the 3D effort which ultimately produced Release 10 (see page 382), work continued on Release 9, aiming for shipment as soon as possible. Upon his return from the first AutoCAD Expo Europe in Birmingham England, where everybody seemed to be asking us for more 3D, Kern Sibbald wrote this memo to propose adding additional 3D capability to Release 9, even if it meant slipping the schedule to do so. His proposal, called "Urgent Worry" (a play on my "Urgent Fury" project which formed a main theme of Release 9), was ignored; Release 9 contained almost no additional 3D functionality.

This significance of this memo lies in its spelling out, in a precise manner, what new features would be required to create a genuine 3D AutoCAD. Kern's specifications here formed the kernel of the 3D feature set implemented in Release 10. In retrospect, we would have been wiser to heed Kern's recommendation of merging the Release 9 and 3D efforts; customers and dealers objected when we shipped Release 9 only 5 months after Version 2.6—they prefer updates no more frequently than once a year. When we shipped Release 10, many users reacted, "thanks for all the 3D stuff, which since I do 2D drafting is totally useless to me," and we had to play catchup on drafting functionality in Release 11. A merged Release 9 and 10, shipped in late 1988 would have been much better received.

Urgent Worry

by Kern Sibbald — April 10, 1987

We have heard a lot about 3D AutoCAD lately. In fact it has become the number one technical priority of the company. This is with good reason, because our users want 3D. Currently, it is probably the most asked for feature in AutoCAD. AutoCAD 2.6 with 3D level 2 will partially satisfy the requests for 3D. My impression from the shows is the users expected more, but they are happy to get some 3D features and are willing to wait a while for more features, but not very long. All of this you have probably heard before and fits with our current strategy of accelerating the 3D project, which I fully support.

At the Birmingham conference in England during the many passionate pleas for additional 3D features, it occurred to me that with AutoCAD 2.6 3D level 2, we have made it clear to our users that we are going to support 3D. And since users clearly want more 3D features and are expecting them, it also occurred to me that if we release an AutoCAD subsequent to 2.6 (White Album)[237] with no new 3D features, we will unwittingly

[237] The code name for Release 9. Amusingly, the fact that the Beatles' "White Album" contained the song "Number Nine" had nothing to do with the choice of code name; we decided to switch from version numbers to release numbers much later, when we couldn't figure out a good version number for this update.

send our users a strong message saying that we not really serious about adding 3D to AutoCAD. Our users want more 3D now, but are willing to wait a little longer. I believe that if we add no new 3D features to White Album, we will seriously erode our what remains of our credibility for creating a 3D product. This could be a very serious mistake. Place yourself in dealer's shoes for a moment and think about what this would mean. On the other hand, if we show progress toward a full 3D by adding some new features, we may be able to retain a neutral position in our dealers and users eyes.

What would the impact on White Album be adding new 3D features at this time? I don't know the answer to that question. But I do feel that it would not seriously detract from the "real" 3D effort if we spend about 4 hours designing a few of the features that we want in the "real" 3D. After that, we can answer the question.

Below I have listed a few of the features that we should consider for immediate implementation in White Album. All these features can be added in a very short time (I estimate one week) if we form a small team of programmers (no more that 5) who will work together to add these features. The time seems right to do it since 2.6 is out the door, and 2.7 has been dropped. My understanding is that although work has begun on the "real" 3D, none of the 2.7 programmers are yet working on it. This would seem to leave a number of programmers itching to do a quick kill project.

Possible New AutoCAD Features:

- Add a new **3DVIEW** dialogue box—or enhance **VPOINT** to include parameters necessary to add perspective and to adjust the view point:

  polar rotation and twist
  look at
  look from
  lens
  film diagonal
  perspective
  view distance

- Make every command accept a 3D point as the first input. If the Z value is subsequently changed (in a 2D command), print a warning message and disregard the Z component.
- Create a "primitive" **CPLANE** command that allows the user to toggle from the current X-Y plane to the X-Z plane, and Y-Z planes for cursor/keyboard input. Similar to ISO toggle.
- Implement a 3D rotate command that drags 3D entities and allows use of the **CPLANE** toggle.
- Add a Z coordinate to the status line.
- Add additional object snap modes in 3D—mid of **3DFACE**, center of **3DFACE**, perp of **3DFACE**, perp to **3DLINE**?
- Implement a Z grid and Z snap that will be activated when the "**CPLANE**" includes the Z direction.
- Eliminate interior faces from polylines, extruded circles, etc.
- Mode to make the **LINE** command behave as the **3DLINE** command does.
- Make new 3D primitives for **BOX**, **SPHERE**, **CYLINDER**, **CONE**, **DISH**. Allow them to be generated about any arbitrary axis.

- Add some way to group or block the **3DFACE**s in **BOX**, **SPHERE**, etc.
- Add a volume clipping dialogue box as in Shade. Normally the volume clipping will be automatically established when perspective is invoked according to the film diagonal specified. However, the user may wish to explicitly control clipping—for example when producing section slices of a drawing.
- Explode a polyline into **3DFACE**s (possibly a mode).

Future Features:

- Add a numeric box. This box could be in the lower right corner of the menu and would contain a numeric keypad with some special characters like backspace, +, -, and return. It would permit the user to enter any number with the use of a mouse or tablet.
- Implement a configurable status line using C printf format and variables names as in a printf command. "**%d %s ...**",**COLOR**, **LAYER**, ... Feed the names into bag as if we are Lisp asking for variables.[238]
- Need a program that will verify that the sizes of dialogue boxes are correct—mostly for the foreign language versions. Could also implement variable size dialogue boxes similar to the alert boxes.[239]

[238]This feature, about the longest-running wish-list item ever, first missed the code cutoff for AutoCAD 1.4 in October 1989 and was finally implemented, eight years and eight AutoCAD releases later, in AutoCAD Release 12 as a MODEMACRO variable which allows DIESEL expressions. In keeping with tradition, this feature was submitted on the day of the code cutoff for Release 12.

[239]This modest suggestion contains the germ of the Proteus Project, which implemented portable, automatically-sized and arranged, user-programmable dialogues. Kern Sibbald designed Proteus and led the development project.

The Golden Age of Engineering

In mid-1987, after we had completed the second public offering, I came to believe that Autodesk should adopt a higher public profile. As the unquestioned market leader in CAD, with close to $100 million in cash, and a market valuation over half a billion dollars, increasing awareness of the company and what it was doing would, I felt, greatly benefit our ability to sell into large corporations and the government.

However, I was afraid that if we unleashed an advertising agency on "corporate communications" we'd end up with something just as bad as all of our previous experiences with ad agencies. Who knows, they might come with something out of a Japanese monster movie, "It is invading your company as you sleep. It is extending its tendrils into your engineering department. It is coming back from the ocean floor and it is mad as Hell!".

But I digress. I wrote this in June to attempt to define an overall communications campaign that we could organise all of our efforts around. I believe that this initial message best sums up the potential Autodesk has in the markets in which it is the leader.

The Golden Age of Engineering

by John Walker — June 19th, 1987

In the lifetime and recent memory of currently practicing engineers a revolution has occurred; a revolution so profound, so widespread, and so rapidly advancing on so many fronts that the enormity of it and its consequences are often unappreciated. But they are real, and they are remaking the world.

The past thirty years have seen an unparalleled advance in our understanding of all of the basic sciences. New materials, such as polymers, titanium, semiconductors, and advanced composites have moved from the laboratory into manufacturing. Microelectronics has grown exponentially since its inception in the 1960's, and has not only made enormously complicated systems possible, as many predicted, it has made them extremely inexpensive, as few expected. This, in turn, has driven the growth of computing technology, placing personal computers in the hands of all who want them, while simultaneously allowing the development of the supercomputers which are becoming key research tools in their own right.

We live in the space age. Since 1962, we have dispatched robots to explore all the major planets, expanding our knowledge from one world to dozens. Men and women routinely fly into space, and space stations are being built by many countries. Our telephone calls and television broadcasts are routinely relayed by satellites a tenth of the way to the moon.

We carry calculators no larger than a credit card that contain more computing power than existed in the world in 1950. We routinely fly to the other side of the globe for a business meeting. And we are thinking about airplanes that fly from San Francisco to Tokyo in 90 minutes, superconducting power distribution systems, fusion power stations, portable telephones that work worldwide, and most of the other stuff of the science fiction of our youth.

Almost without noticing, we have entered an era where the fundamental question is not "What *can* be done" but "What *should* be done".

Truly, this is the golden age of engineering.

But even more, it is a golden age for the individual engineer. Driven by technology, design is not dominated by the all-encompassing government design bureaus many imagined in the 1930's, nor by an oligopoly of giant companies as many saw in the 1960's. Instead, the basic tools to invent, design, and manufacture have become so inexpensive and widespread that "downsizing" has become at least as much an imperative in management as in design.

We are entering the age where we are limited primarily by our creativity. Our ability to imagine, and the courage to make our dreams into reality will be our most precious resource. In this age, the designer has a resource that most designers of the past could hardly imagine—the computer. Engineers who, less than twenty years ago, toiled into the night with log tables, slide rule, and pencils, making parts, then breaking them on testing machines, or designing circuits and struggling to get them to work can now design on their desktops with productivity hundreds to thousands of times greater. And the products of their minds in turn accelerate the process.

Anybody who attempts to predict what we can do in our lifetimes should first reread predictions made in 1960. Anybody who draws a limit to what our children can achieve is a fool.

Autodesk designs, develops, manufactures, sells, and supports key computer-aided-design tools. We are working as hard as we can to make them worthy of the tasks to which they are put by the designers of this golden age.

Autodesk, Inc.
Sausalito, California

Tools for the golden age of engineering.[tm]

AutoCAD • The Engineer Works • CAD/camera • AutoSketch • AutoShade
AutoCAD AEC Architectural • AutoCAD AEC Mechanical

Cosmic Perspective

Nobody can resist the temptation to ask, every now and then, "what does it all mean"? At least I can't.

Cosmic Perspective

At attempt to gain perspective in various domains at several different orders of magnitude. In three parts.

by John Walker
June 29th, 1987

Cosmic Perspective—A

Autodesk recently announced the shipment of the 100,000th copy of AutoCAD. Let's do some calculations of the size of the industry this represents. Throughout this paper, the numbers I calculate will be expressed in "astronomical units", precise to 10 decimal places but probably accurate to 1 or 2.

Since Autodesk's sales curve is reasonably approximated by an exponential, it's reasonable to assume an average retail price for AutoCAD of about $2,000, factoring in discounting at retail, earlier sales at lower prices, and sales of base, ADE-1, and ADE-2, all corrected for sales of foreign language versions which carry a premium over the English version and have recently benefited by the fall of the dollar.

This means that, at retail, total sales of AutoCAD to date are about $200 million. But we just sell the software. Charles Foundyller estimates that software accounts for about 15% of the revenue in the PC CAD business. If this is accurate, the total retail sales generated to date by AutoCAD is approximately $1.3 billion. If you discount the obscene wealth extracted from this market by a few people, assume that half the business has been done in the last year, assume an industry-wide living wage of $40,000 per year (remember overhead), and attribute a negligible materials cost to AutoCAD-related products, this means that roughly 15,000 people earn their living from the AutoCAD industry (this is, of course, an abstraction for a much larger number of people partially supported by the industry).

None of this existed in 1982.

Cosmic Perspective—B

An instrumentality of the federal government of the United States of America bought the 100,000th copy of AutoCAD. In its 1987 fiscal year, the federal government spent $1.3 billion, equal to the total five-year market for AutoCAD related products, about every twelve hours.

Roughly half of Autodesk's profits during the last five years have been paid as taxes.

Cosmic Perspective—C

At 07:35:35 UTC on February 23, 1987, neutrinos and photons from the exploding star Sanduleak −69°202 reached the Earth. Roughly 10 billion neutrinos from the supernova passed through every square centimetre of the Earth's surface. In approximately five seconds, 10^{58} neutrinos were emitted—equivalent to the total conversion into energy of one tenth of the mass of the Sun.[240]

Over 99% of the energy of the supernova was carried away by the neutrinos; the visible manifestation in the sky is caused by much less than 1% of the energy released. The energy emitted in five seconds by the supernova is roughly equal to the output of the Milky Way galaxy for a period of several years. So great was the neutrino flux that, despite the fact a beam of neutrinos is attenuated only 50% by passing through six light-years of lead, approximately one million people on Earth experienced a neutrino interaction in their bodies as a result of the supernova.[241]

The star that exploded is 160,000 light-years from Earth.

[240] A survey of SN 1987A results in *Science*, Vol. 240, Page 754, offered additional perspective on the neutrino burst. The neutrino luminosity in the first second was 10^{53} erg/sec. Using consensus numbers for the density of luminous matter and the size of the observable universe, the luminosity of the *entire universe* is approximately 5×10^{52} erg/sec. Thus, during the explosion the supernova was brighter than the entire rest of the universe, yet generated all of its energy in a region about 50 km across. By comparison, if the Sun shines for another 10 billion years its total energy output over that entire period will be about 10^{51} erg. The supernova released 100 times that energy *each second* of the collapse.

[241] The neutrino flux through the Earth was 50 billion neutrinos per square centimetre.

External Tanks

Many aerospace experts believe that the most cost-effective way to build a large orbital facility is to convert the External Tanks currently discarded on every Space Shuttle mission into habitable space. External Tanks Corporation, 80% owned by a consortium of universities and staffed and advised by a distinguished group of space statesmen and Washington hands, was first to do the extensive groundwork to bring this about.

When I learned they were seeking funding, it seemed to me a natural opportunity for Autodesk. Here was a group trying to do with the very highest of high technology, manned space operations, precisely what Autodesk had done to CAD—reduce the cost by one or two orders of magnitude and thereby increase the market size by an even larger factor.

In addition, the External Tank engineering project seemed a natural to apply Autodesk's engineering tools in environments outside the traditional, heavily funded, aerospace sector.

When we made the investment in External Tanks in July of 1987, negotiations were underway with NASA. Since then, commercialisation of the External Tank has become part of the official U.S. national space policy, External Tanks has received an official NASA Memorandum of Understanding to develop the tanks, and NASA has undertaken preliminary work to study External Tanks Corporation's proposal to convert the tank into a Gamma Ray Imaging Telescope. This is the second draft of the press release, expressing our joint intent; the actual press release sent out was watered down somewhat to avoid offending certain parties.

Autodesk Invests in External Tanks Corporation

Sausalito, California, July 8, 1987.

Autodesk, Inc. and External Tanks Corporation jointly announced today that Autodesk will purchase approximately 5% of the common stock of External Tanks Corporation for $225,000.

External Tanks Corporation's aim is the development of an orbiting research and manufacturing facility called Labitat[tm], based on the External Tank (ET) which is launched as part of every Space Transportation System (Space Shuttle) mission and is currently discarded. External Tanks Corporation is 80% owned by the University Consortium for Atmospheric Research (UCAR), a group of 57 universities who have administered research programs and facilities for over 25 years.

NASA is presently in the final stages of negotiating a Memorandum of Understanding with UCAR and External Tanks Corporation under which title for tanks lifted into orbit by STS missions will be transferred to UCAR

and administered by External Tanks Corporation. External Tanks Corporation will transform these tanks into orbiting facilities which will then be leased to government, academic, and commercial customers at prices which are potentially one thousandth that of comparable orbiting facilities.

Alvar Green, President of Autodesk, Inc. said, “The phenomenal success of Autodesk has been based on taking computer aided design technology, which was previously thought to be priced out of the reach of all but a very few large users, and making it available to millions of people at costs a fraction of what was previously imagined. We hope, through our investment in External Tanks Corporation, to spark a similar reduction in the price of space research, observation, and manufacturing. We believe that as the computer revolution has shown and as the success of AutoCAD has demonstrated, every time you radically reduce the price of a product, a far, far larger market emerges. External Tanks Corporation can do for space research and development what silicon technology has done in the computer industry, and we’re positioning ourselves to promote this and reap the benefits as it happens. We’re also looking forward to applying our expertise as the world leader in computer aided design, and the pioneer in bringing solid modeling to the desktop to this challenging venture. Our participation in this venture will help demonstrate PC-based design tools in a space age, man rated project.”

Dr. Randolph Ware, President of External Tanks Corporation said, “As the United States strives to regain the momentum of its space program and resume its leadership in space science and commercialization, the need for a low cost, large volume research and manufacturing facility in space will become manifest. As the National Commission on Space has recommended and numerous previous studies have shown, commercialization of the External Tank provides the lowest cost, nearest term solution. At a time of growing concern over our space program, NASA’s provision of the previously discarded External Tank resource via a Space Grant to UCAR will parallel the Land Grant program of the 1800s in paving the way for research, development, and near-term industrial opportunities on the unlimited frontier of space. We’re proud that Autodesk, Inc. will be joining us in developing and exploiting this enormous opportunity.”

Source Distribution

Developing a program as large as AutoCAD on personal computers, with most programmers working off-site, presents some interesting logistical problems—mainly, how do you distribute the source code which presently occupies over twenty megabytes. The source has outgrown every medium we have chosen, and at times we've even handed out paper bags with twenty or more floppy discs.

We finally settled on Iomega Bernoulli Boxes, with twenty megabytes per cartridge. Naturally, the source grew to fill them. This is a modest proposal to buy some time before the inevitable happened. It wasn't implemented.

Combining the files in a source distribution into a composite archive file can enormously speed up the process of copying source distribution media. Benchmarks, recommendations, and a moral are presented.

by Kelvin R. Throop
September 29th, 1987

On the morning of July 18th, Liberty, New Hampshire became the first town to vanish. Many residents of New Providence, over a wooded ridge from Liberty, were awakened at about 3:30 A.M. by a clap of thunder. Those who looked outside saw clear sky and a glow in the direction of Liberty. Volunteer firemen called friends in Liberty to ask what had happened, but none of the calls were answered. Most people went back to sleep.

By midday, all the world knew what had happened, but nobody knew how or why. The town of Liberty was *gone*. Gone to a meter below the ground. Gone right to the town limits, where some branches had fallen on undisturbed grass after their trees had vanished.

•••

There has never been a truly satisfactory way of distributing full source releases of AutoCAD. The product has grown so rapidly that it rapidly outgrew each medium selected for distribution and became a large burden to copy and maintain.

Largely at my urging, Autodesk spent a large sum of money to equip all AutoCAD developers with Iomega Bernoulli 20 megabyte drives and has purchased an seemingly endless supply of cartridges for these drives. On the face of it, the Bernoulli is an ideal source distribution medium.

- It is fast. Average access time is comparable to a high performance Winchester hard disc drive.
- It supports a true DOS file system. Files can be extracted with normal DOS commands. No special archiving tools are required.
- It is a high-density medium. Each cartridge stores over 20 megabytes formatted. This is enough for an entire AutoCAD source and object distribution.

In practice, several disadvantages have become apparent.

- They don't work very well. Cartridges which verify one day may prove unreadable the next.
- Software support is shaky. Several bugs which can cause loss of *all* data on a cartridge have only recently been fixed.
- Image copy is untrustworthy. Binary image copy from one cartridge to another seems to frequently result in undetected data errors. There is no confidence that image copy correctly intercalates the bad track maps of the cartridges involved.
- The alternative, MS-DOS copy of entire directory trees, is painfully slow.

But hey, what do you want, it's from Utah, right?

Right.

• • •

The president appointed an investigating panel chaired by the secretary of defense and made up of the secretary of the interior, the chairman of the National Academy of Sciences, and the president of M.I.T. On the 20th, the group held a press conference in Manchester and announced that no probable cause had yet been found. The defense secretary said that hostile action had been ruled out "for the time being", since no aircraft were in the area at the time, nor were any satellites tracked by NORAD overhead. "In any case", the secretary concluded, "we possess no technology which could do this, and we don't believe our adversaries do either".

• • •

Well, I got us into this, so I decided to spend some time seeing if I could get us out. The first thing I did is make some measurements of the AutoCAD **`X.0.60`** (8/2/87) full distribution Bernoulli. I copied the directory tree from this cartridge to a Unix file system over NFS. I then made a **`tar`** file of the entire directory tree copied from the Bernoulli. The total size of the data on the Bernoulli was 19,786,240 bytes.

All of the measurements presented herein were made on an IBM PC/AT, 6 Mhz version, with an Iomega 2010 or 2020 removable cartridge disc system. In all of the following timings, I assume that the Bernoulli cartridges being written have been previously formatted. Formatting takes approximately 4 minutes per cartridge. Since all operations require a formatted cartridge, this is invariant under the options we're exploring and can be added to all the numbers presented below.

• • •

Over the next week ten more towns vanished: two in Massachusetts, another in New Hampshire, three in California (including one suburb of San Francisco), one in New Mexico, two in England, and one in the Netherlands. Data began to accumulate about the phenomenon. One of the California towns vanished during a Landsat pass; the multispectral camera recorded only the glow of ionized air molecules recombining. The nuclear test detectors on the remaining Vela satellite and the monitors on the Navstar constellation observed four light flashes coincident with disappearances. Nothing like the double flash of a nuclear detonation was seen, just slow airglow decay. No prompt radiation was detected at the time of the disappearances, nor was residual radiation found at the sites. Electromagnetic transients similar to a very large lightning strike were detected, and underground solar neutrino experiments reported six neutrino events near the time of the flashes, but gave only a 60% chance that this was correlated. *Aviation Week* reported that some at Los Alamos believed the flash spectrum similar to a free electron laser, but they had no idea how this could occur spontaneously.

• • •

First I measured the time to perform a file-by-file copy of all files on the distribution using Metaware's **`FIND`**. The file by file copy from one Bernoulli cartridge to another took a mind-numbing 63 minutes: one hour and 3 minutes!

The discrepancy between this and the time required to simply transfer the data from one cartridge to the next was elucidated by performing an image copy from one cartridge to another. The Iomega **`RCD`** utility copied the entire cartridge in a tad less than 5 minutes: more than twelve times faster.

• • •

As in time of war or natural disaster, the population surprised the politicians with its equanimity. Certainly there was uneasiness, and frustration grew as days passed without any explanation or plans to deal with the crisis, but no real signs of panic emerged. If scientists had no theories (as one physicist put it, "nothing even deserving of the term wild guess"), explanations nonetheless abounded. Television evangelists seized on the crisis as demonstrating God's wrath on sinful man (though none understood why Las Vegas was still around). The *National Star* interviewed 75 prophets and psychics who had predicted the disappearances, but was silent on which cities the "UFO Aliens" would kidnap next. Sinister rumors of Soviet secret weapons circulated, supported by the fact that no Eastern Bloc city had vanished.

• • •

Clearly, something odd is going on here. While one expects a hard-coded device-specific image copy utility to run faster than the operating system's copy facility, a factor of two is more than one typically gains. But *twelve times faster?* Hmmm. . . .

Next I decided to try copying the 19.7 megabyte **`tar`** file I made from the distribution onto a Bernoulli across NFS. The entire copy operation took 6.4 minutes. Note that this was a DOS copy across an Ethernet link from a Unix file system, yet it was only 28% slower than Iomega's much-vaunted image copy facility. As all of those who have studied at the feet of the legendary masters of gonzo programming (especially those who did their studying in Cleveland) know, factors of ten *percent* may stem from sloppiness, but factors of ten invariably indicate idiocy. It was clear that somewhere deep within the sanctum sanctorum, the very nucleus of the operating system, there was some really major league *evil*.

• • •

By September 1st, over one hundred villages, towns, and cities in the United States, Western Europe, Latin America, Japan, and Australia had evaporated into the dead of night and the world was beginning to go truly crazy. Not one Soviet or Eastern European town had been affected; NATO moved to alert status "as a precautionary measure". Still, no pattern emerged. Mostly small and medium sized towns and suburbs were vanishing. In the U.S. most were on the East and West coasts. Most of the mid-continent disappearances were university towns. No large cities nor unincorporated areas had yet gone, and people began to flow to the cities. Squatter camps appeared in state and national parks.

• • •

So, back into the Honda and down the hill to Duff's machine (which, unlike my humble configuration, has a gen-u-wine two-holer Bernoulli). I took the Bernoulli onto which I had copied the 19.7 megabyte **`tar`** file and tried copying it to an empty cartridge with a simple DOS **`COPY`** command. The entire copy completed in 5 minutes and 43 seconds.

At this point it's worth recapping the timings in these experiments. All timings in this table are in seconds.

Function	**Time**
Format cartridge.	240
Metaware **`FIND`** copy entire cartridge file by file.	3815
Iomega **`RCD`** image copy entire cartridge.	298
Copy **`tar`** file over NFS to cartridge.	386
Copy **`tar`** file cartridge to cartridge.	343

What can we conclude? Clearly the enormous difference between the time required to copy the **`tar`** file and the time to reproduce the entire file structure on the target Bernoulli is simply the time that the operating system required to create all of the directories and files in the source distribution. Since the inefficiency is in the nucleus of the operating system itself, there is only one way to get around it.

• • •

For perhaps the very first time, a librarian came to the rescue of civilization. Todd Murphy was a researcher for the Library of Congress project to build a computer database on the vanishing towns in the hope of finding some common thread or pattern. But the brain is still the best computer when it comes to finding patterns. The answer came not from the database, but to Murphy's mind as he was entering data in the middle of the night.

• • •

Clearly, since the problem is in the operating system, the only way to overcome it is to bypass the operating system. Hence, we should prepare our source distributions as **`tar`** archives which can be copied more than

ten times faster than fully elaborated MS-DOS directory structures. Fortunately, the Metaware **FIND** utility (licensed to all AutoCAD developers as part of the High C distribution) can write Unix-compatible **tar** files. Using Unix **tar** format allows Unix systems to process the distribution without format conversion. If the current directory contains a complete AutoCAD source distribution, you can prepare a Unix **tar** format Bernoulli on drive **L** with file name **DIST.TAR** with the command:

```
find *.* -utarc L:DIST.TAR -t
```

Having prepared a Bernoulli containing a **tar** distribution, you can extract the entire directory tree into the current directory with the command:

```
find -utarx L:DIST.TAR -cp .
```

If you have access to a Unix system, you can copy the **tar** file from the Bernoulli to Unix as a single binary file and extract the component files under Unix with the command:

tar xfv *tarfile*

where *tarfile* is the name of the file you've copied the entire distribution **tar** file into. Note that files are archived as they were originally stored; no end of line convention conversion is performed. Hence, even if you de-**tar** the archive on Unix, you will end up with source files in MS-DOS format.

• • •

That Murphy is forceful and persuasive as well as wise was evident when, after hurriedly checking his hypothesis against the list of cities and finding complete confirmation, he convinced the White House switchboard to awaken the vice president and the national security adviser and ended up at 4 A.M. declaiming to a small, bleary-eyed group in the Situation Room, "Congress has to act on federal overriding legislation today, and you must get the President back in town to sign it before we lose any more people. Get State to work contacting the Europeans right now—it's going to be night soon over there. And *call the prime minister of New Zealand!* Every town that vanished had declared itself a nuclear-free zone. And the nuclei are moving out."[242]

• • •

So, Horatio, the problem lies not in the stars nor in Roy, Utah, but rather in the nucleus of MS-DOS. Consequently, only one solution is possible. We must move to a completely *nuclear-free* software distribution format—one which can be replicated without a single call to the inefficient heart of MS-DOS. The **tar** file mechanism suggested herein provides such a solution and allows creation of AutoCAD full distribution media more than ten times faster than previous techniques without compromising data integrity.

I also investigated compression techniques. Our distribution almost fills a cartridge at present, and already excludes some items which should probably included in a full distribution (e.g., the Kelvinator). My experiments indicated that compression can help us fit far more data on a cartridge (and commensurately decrease the effective time to copy the original, uncompressed, data), but the convenience costs of compression are fairly high. I

[242]Not long before I wrote this chapter, Mill Valley California, my home town at the time, declared itself a "nuclear weapons-free zone." This was devastating to the local economy, forcing closure of Mill Valley's three plutonium reprocessing plants, the forty Minuteman silos, and the secret Trident submarine pens lurking under the physical ægis of Mount Tamalpais. Mill Valley's subsequent transformation into a kitschy tourist trap became inevitable.

started with the original 19.7 megabyte composite **tar** file and used the Unix **compress** program. This reduced the file size to 8,751,305 bytes, saving over 55%, but compressing this file required almost 9 minutes of elapsed time on a Sun 3/260. Since the Unix compress program runs only on Unix, and Unix cannot directly read a Bernoulli, compressing the entire distribution in this manner means one must copy the entire 8.7 megabyte compressed **tar** file to Unix, then decompress it, then extract files from the archive. At the instant that the compression or decompression is complete, the entire mess occupies over 28.5 megabytes on the Unix system—a very severe free space constraint on any system. Decompressing the file takes about four minutes of elapsed time.

Since a large part of the problems with the Unix compress program stem from its inability to run on MS-DOS, I tried Kern's **fsq** program, which runs on both Unix and DOS. I was about to glibly say, "and it's still running", but it just finished. I don't think we'll use **fsq**: it took 39 minutes of elapsed time to compress the composite **tar** file on the Sun 3/260 (27 minutes of CPU time), and it only reduced the file to 17,640,789 bytes, saving less than 11%.

Compression seems impractical when applied wholesale to the entire distribution, but can be useful for compressing smaller parts of the release. Duff has suggested that we **tar** the files directory-by-directory and compress at that level. This would drastically reduce the disc space required to decompress each part and only marginally increase the time needed to copy the cartridge.

Finally, I have been thinking about how to guard against undetected errors in these distributions, however copied. Neither Unix **tar** nor Metaware **FIND** in its native archive mode provide any form of file checksum. I am willing to write an external checksum utility to guarantee accurate distributions. If the following description seems a tool we will actually use, I will invest the day or so to write it.

To make a distribution, compress the directories of the distribution into one or more **tar** archives. After creating the archives on a trusted system and compressing them if you like, run the checksum program on each archive:

feathers -m *tarfile*

This program reads the named *tarfile* and checksums each block in the file with a highly reliably checksum. It creates a checksum file named *tarfile***.cks**, and reversibly encrypts the first 128 bytes of the *tarfile* so that **tar** cannot unpack the file. All of the **tar** files and their corresponding **.cks** files are placed on a master Bernoulli and then copied for all recipients.

When a developer receives a Bernoulli, he verifies all of the files on it with the call:

feathers *tarfile*

This validates the file against the corresponding **.cks** file and, *only if it is correct*, decrypts the **tar** file so it may be de-archived. This absolutely guarantees that no developer can end up with corrupted data (unless an error sneaks past the checksum algorithm).

Nuclear Free

The Morning After

On October 19th, 1987, the stock market crashed. The Dow Jones Industrial Average plummeted more than 500 points, almost five times the previous record single-day drop (which occurred the previous week), and in percentage terms, almost twice the size of the Crash of 1929. This was part of, and in turn accelerated, a global collapse in equity markets. Autodesk's stock, which had hit a high of over 33 less than three weeks before, plunged to close below 13 on October 27th. This decline wiped out all the gains in the stock for the last year, and left almost everybody hired in the last year with above-market stock options.

In addition, dire predictions about the future, concern for the value of retirement plan investments, and business prospects in general contributed to a general climate of unease. I made the following remarks at a special company meeting on October 27th.

Remarks for the Stock Market Crash Meeting

by John Walker — October 27th, 1987

Welcome to the morning after.

It's hard to watch frenzy in the stock market, 500 point drops on the Dow Jones Industrials, Autodesk stock shedding almost 60% of its value in less than a month, and ponderous predictions of panic from pompous prognosticators without wondering, "what does this mean to me, and what effect will this have on our company".

I'll try to explain what's going on as I see it, the steps we took months ago to prepare for this inevitable occurrence, and what we can expect to happen next. Then I'll be glad to field any questions you may have on any subjects except the three I never discuss: politics, religion, and text editors.

I'd like to read a quote from a memo I wrote on February 3rd of this year:

> I'm beginning to get nervous, *really* nervous, about this stock market runup and the action in our stock.
>
> I think we may be entering into the kind of wild speculative blowoff that ran technology stocks into multiples of a hundred times earnings in the 1960's.
>
> And you know what happens on the morning after.

As I recall, I started worrying at about the point the stock passed 13 (correcting for the split) on the way up. As one who has seen the crash of ’70 and the crash of ’74, it was abundantly clear to me that we were in one of those rampaging bull markets in which common sense, fundamental values, and historical perspective are trampled in the wild rush to get rich quick.

This happens every decade or so, and every time it happens, there invariably comes a time when all the news is rosy, when business has never been better, and when every last person in the world who is inclined to buy has bought. Then the bottom drops out.

Why? Because prices rise when everybody wants to buy and they drop when everybody wants to sell. But when everybody has already bought, there aren’t any buyers left. Then somebody looks around and notices that it’s a long way down, and that those black clouds that everybody ignored last week are now casting cold shadows and beginning to drizzle.

Bull markets lead to bear markets lead to bull markets in a cycle as old as commerce and as inevitable as the changing of the seasons.

So what does the stock market crash mean to Autodesk? In terms of direct effects, essentially nothing. The market value of our company has been cut in half, but since the company has no plans to sell additional stock, that has no effect on our operation. Since Autodesk’s stock has gone down about the same amount as other comparable companies, we’re in the same relative position as we were before. A receding tide deposits all boats on the muddy bottom.

Autodesk’s business is as good as it’s ever been. Our sales are stronger than ever, we continue to be one of the most profitable companies in the world, and we’re expanding our product line, our distribution channels, our presence in major accounts and government markets, and our international operations. None of this is affected in any way by gyrations in the financial markets. When the value of our stock doubled in the last year for no good reason, it didn’t make our business grow any faster than we had predicted two years before. And when our stock falls by one half, there isn’t any reason to think that will affect us either.

Now none of this is to minimise the impact of what is going on. When you hack a trillion dollars off the value of financial assets, it’s bound to have some impact on business and the economy. And if we find ourselves in a deep recession, our sales may be hurt. But amid all the comparisons with 1929, let’s remember that not every stock market crash leads to a recession. The stock market crash of 1929 was only part of the great depression, which devastated the economy primarily due to the collapse of the banking system, not the stock market. That type of collapse is much less likely to happen today. This may be more like the crash of 1962, where the stock market hit a speed bump and scared everybody to death, but there was almost no effect on the economy, which continued to roar ahead and eventually carried the stock market to record highs in 1966. We don’t know what will happen this time, but we have carefully prepared for whatever the future may bring.

If you’re inclined to worry about the future of Autodesk, keep this in mind. We have one hundred million dollars in cash. That cash is invested in risk-free short-term government securities whose value is not affected at all by the stock market and whose value could be at risk only if money became obsolete. One hundred million dollars. That’s about five dollars per share of stock, and I sometimes wonder if some of the people who are dumping our stock at 13 know that there’s a green five-spot behind every share. One hundred million dollars. That’s about three hundred thousand dollars for every employee, so even if sales went to zero, we have a multi-year buffer to keep the company operating and progressing. Autodesk is one of the most financially strong companies around. We got that way by deliberately choosing a prudent and conservative path to growth, politely declining to play the financial games that got so many in trouble in the recent turbulence.

We have no debt at all. We do so much business overseas that when the dollar falls, it actually improves our sales and profits. If the United States goes into a downturn, we'll do plenty of business in the rest of the world while we wait for better times. We don't gamble the future of our company in the stock market or in other short term speculations. We're building this company to be a force in the market for twenty, fifty, a hundred years to come—and we're not going to crapshoot it for a quick buck.

But most of all, what we are doing is *right*. We are not just a good stock, we are a good business. We sell tools that individuals and companies use to make themselves more creative and productive. And productivity is more important than ever when the economy turns down.

A month ago, I wrote the copy for advertisement which I now wish we'd been running the last two weeks. It's only two sentences, and I'd like to read it to you now.

> "Everybody says America must increase its productivity or reduce its standard of living."
>
> "Autodesk sells products that increase the productivity of every manufacturer in America."

It's true, you know. And people who work hard, in companies that work well, who make and sell products that make a difference, that make people's jobs easier, their lives more productive, and enable them to do things they couldn't do before, do well regardless of the stock market, the economy, or the folly of politicians.

I have called this "The Golden Age of Engineering".[243] Part of that golden age is a resurgence in interest in truly productive endeavours, designing, building, and manufacturing, rather than corporate takeovers, leveraged buyouts, stock index futures, and "financial services".

Not long ago half the pundits in the world were explaining how Americans could prosper by selling innovative financial products to one other while importing productive work from Europe and the Far East. Personally, I've always thought that was a lot of turbo. You know, turbo, that's a device that makes hot air and high pressure. Now, I think, people are awakening to the folly of this monumentally silly idea. And as they focus on productive work and tools for that work, they will continue to buy those tools from the world's leading supplier, Autodesk, and the company that we have built will continue to prosper.

• • •

All of this may seem small consolation if you own Autodesk stock and have watched its value plummet by one half, or if you have stock options whose exercise price is now well above the market price of the stock. For option holders, things are not as bad as they may seem. When I wind up in a minute or so I'm going to turn things over to Chris Record. He'll describe the Amazing Counter Parabolic New Options For Old Trade In Plan. Take note, it's very important.

If you're a stockholder, there's nothing for it but to come to terms with the fact that the stock isn't worth what it was two weeks ago. If you were looking at that stock as the down payment on a house, the kids' college fund, or some other economic goal, the adjustment can be tough to make. But it's important that we don't let regret paralyse us at a time when our efforts are vital to keeping the company growing and eventually restoring the value of the stock as investors come to their senses and realise that Autodesk is different from the many other companies they dumped in time of panic.

[243] See page 392.

You've all heard me talk again and again about the fact that this is a high-technology, high-risk business and that great rewards flow from the assumption of large risks. For the last five years, and especially in the last year, we've all shared in those great rewards. Now we get to taste the risks. It's both gratifying and frustrating to have your stock dumped when you didn't do anything wrong. Most small companies get pummelled when they fail to get a product to market on time or they can't meet their sales forecasts. We do everything right, and still we get bashed. It's little comfort indeed to know that everybody else is getting clobbered also.

We may look back on this period as a time of trial from which much good flowed, a time which tempered the company for its next great surge of success. Good times are wonderful, but they can lead to a feeling of complacency which can sow the seeds of destruction for any company. As I've said repeatedly, Autodesk is just one small company in a dangerous, highly competitive world. If this gust of winter's chill makes us rededicate ourselves to making our products, our service, and our marketing and sales the very best in the world, then I continue to believe what I said at the annual meeting in 1986.

> Autodesk has always competed like a hungry rat. We will continue. And we will prevail.

Glasnost

Throughout 1987 I had been becoming increasingly convinced that Autodesk's long-standing policy of tight security regarding new product releases was growing counterproductive. My experience with Univac mainframes, where there was essentially full disclosure to the customers of Univac's software development plans, persuaded me that such an environment was far more conducive to developing products that met the customers' actual needs.

The question always was, "can we afford the financial risk of disclosure followed by a late shipment, and the competitive risk of laying out our development plans for all to see?". In this memo, I argue that we can. Pursuant to this proposal, for the first time we briefed our dealers and developers on the features to be announced in AutoCAD Release 10 (Abbey Road), well in advance of its official unveiling at AutoCAD Expo in May of 1988.

To: Al Green, Dan Drake, Malcolm Davies, Chris Record, Eric Lyons
From: John Walker
Date: 20th February 1988

Subject: Гласност

For some time I've felt that our policy of high security with regard to the features to be included in future releases of AutoCAD and general information concerning their release schedule, was growing increasingly outmoded in light of our dominance of the market and our desire to maintain a close cooperative relationship with our resellers and their customers. In this memorandum, I'd like to recap the reasons that caused us to originally adopt our policy of secrecy and then examine the changes in the competitive environment which have, in my opinion, caused this policy to become not only unnecessary, but actually damaging to our competitive position.

In this document, I will present the case for changing our policy to one of essentially complete openness and full disclosure regarding our plans for the future development of AutoCAD; moving toward a cooperative relationship with our user community more like that between a mainframe computer vendor and its user group, or MacNeal-Schwendler and the NASTRAN user group. I will present this case in the manner in which I communicate best: as an advocate. The fact that I present a strong argument here should not be taken as an indication that I harbour no doubts about the possibility of this policy backfiring, or that I am unaware of many equally strong arguments for continuing our current policy. My purpose in presenting this case is to make us reconsider what we are trying to accomplish with our current policy and then rationally decide what changes may be in order, instead of blindly maintaining the status quo.

The original rationale for maintaining security when a new release of AutoCAD impended was simple survival of the company. Since we never offered any formal inventory protection to our resellers, and because each

release of AutoCAD contained a large collection of additional features, news of an upcoming release would cause dealers to defer orders until after the release became available. On several occasions when news of a forthcoming release leaked out, we saw our sales drop to half their previous level for the one or two months before shipment of the new version.

Since the company was so thinly capitalised, premature announcement of a new product followed by a shipment delay, for whatever reason, could place the company at risk of bankruptcy. Consequently, the prudent course was to maintain tight security about the content and schedule of upcoming releases. We even went so far as to try to eliminate notification clauses from OEM agreements we negotiated. Obviously, we had no desire to create ill-will by loading up resellers with soon-to-be obsolete product, but neither could we run the risk of sinking our company.

After the 1985 initial public offering, we had enough money that order deferral couldn't bankrupt us, but another imperative asserted itself—the need to report quarterly increases in sales and profits. Since we had experienced one- to two-month 50% sales fall-offs, followed by compensating sales bulges after shipment, it became essential that if one of these was to occur, it be totally contained within a fiscal quarter. Otherwise, the Wall Street isomorph of wind shear could buffet Autodesk's standing in the financial community.

So, our policy was total secrecy about what we were working on, and "no comment" regarding shipment dates. This policy was the product of the environment in which we operated at the time, and served us well. But I believe that it may have now become outdated, and may actually be working to our detriment.

At the time we adopted the policy of secrecy, we were a small, virtually unknown company attempting to establish itself in a market which was largely unrecognised by mainstream microcomputer analysts. Today, we command over 60% market share in a rapidly-growing market which is viewed as one of the primary applications for the high end of the desktop computer market. Autodesk's installed customer base dwarfs any competitor, our distribution network essentially controls the route of engineering software from producer to customer, and we have encouraged the development of a burgeoning market in third party software, hardware drivers, education, books, and consulting. Our goal was to become an industry standard, and I believe that we have achieved that goal. Now, I'd argue that it's time to start acting like an industry standard rather than a marginal producer, and that so doing can reinforce our position as the standard.

But enough generalities: let's look at AutoCAD Abbey Road, and examine the consequences of the two approaches on our market. First, it's worth noting that Autodesk took a totally unprecedented step when we announced, concurrent with the launch of Release 9, that Release 10 would contain full three dimensional capability and would be shipped in the first half of 1988. Never before had we so explicitly called our shots regarding the feature content of a release and its shipment date. The reasons we did this were obvious—we'd been assailed by every competitor (even those without a product, or those with somebody's else) for "not having 3D", and after a long hiatus during which we had no marketing presence whatsoever, we felt we had to respond with an explicit statement of when we would "have 3D".

This step was actually part of a series of liberalisations of our product secrecy which had occurred over the preceding two years. For releases 2.5, 2.6, and 9 we had formally briefed third party software vendors prior to the announcement, and given them pre-release copies of the product to adapt their products to. We did this even though many third party developers are also dealers, and hence could adjust their own ordering schedule based on the knowledge of the new release (yes, we never said when, but when you call in the developers 60 days before AutoCAD Expo, it isn't very hard to guess the launch date). In several OEM contracts we agreed to disclose new release dates in advance. Prior to the release of 2.6, we described the product in advance and demonstrated it to industry analysts. Since the White Album announcement, we've shown Abbey Road to

selected subsets of our constituencies based on our feelings for the market. We've shown several customers in-depth previews of the product, and we've used their reactions to guide our development of features for inclusion in the final product. We've briefed book publishers about the features in Abbey Road. And most recently, of course, we have held a very successful pre-release disclosure to our ADI developers, notwithstanding the fact that many of these developers also support competitive CAD products.

What, precisely, are we gaining by withholding the details of what we are doing in Abbey Road from our general user community? I believe that the expectations for Abbey Road are much lower than the reality. I don't think that anybody really expects that we'll deliver multiple on-screen views, smooth surfaces, and database handles in addition to the already announced generalisation of AutoCAD to three dimensions. The total impact of the seamless growth of AutoCAD to 3D—the product of Scott's[244] genius—cannot be appreciated until seen, or better yet, used. If we're in a competitive environment where we've allowed our competitors to create the impression that they have 3D and we don't, then why in the world do we hide the wonder of our 3D release under a bushel? The mere announcement of the feature set for Abbey Road, which could consist of a pre-release of the manual supplement working document would, it seems to me, erect an insurmountable barrier in front of any competitor who based his sell on "AutoCAD lacks 3D". We've already said what, 3D; we've already said when, first half. Why clam up on the details when those details will end all uncertainty about our fully supporting 3D?

Now let's step back from the details of Abbey Road to the more general question of disclosure of our future plans for AutoCAD. And it's AutoCAD I'm talking about—I see nothing to be gained and much to be lost by talking about new products in advance. But if we really believe that AutoCAD has become an industry standard, almost by definition unassailable as long as its vendor maintains a technological lead, then shouldn't we make overt the partnership between the vendor, reseller, customer, and third-party value-adder community which is the essence of a de facto standard?

In one stroke, we'd be moving from the inherently adversarial relationship of those who know with those who must guess to a partnership—a partnership forged from common interest in the future of AutoCAD: those who make it, and those who have bet their companies on it. I think it would lift a veil of implicit conflict almost as obscurant as that of the hardware lock, and with no more consequences on our financial results than our expunging that particular bleeding sore. It would also be a brassy statement of self-confidence mixed with humility: we'll tell you where we're planning to go, and we're listening to you to hear if we're solving the problems you feel are important.

It would also give third party developers a clear message on what areas were safe from Autodesk's future development, and which areas had only a limited window of opportunity before Autodesk entered them. At the technical level, knowing the design of some forthcoming features might be very helpful to a developer in the midst of his own long-term project. For example, knowing how we're doing database handles would guide those who we expect to use them. This communication would not just benefit the developers, however. At every single developer briefing, we discover some oversight in what we've done that would have made the feature much more useful, but which it's too late to change and maintain the shipping schedule. Open, two-way communication would let us remedy these shortcomings before it was too late. After all, the very best guidance on the design of a product comes from the people who are actually using it, and by developing behind a wall, we insulate ourselves from much of the interchange of ideas which could help us provide the best solutions.

From a political standpoint (using "political" in the sense I define it "the means by which groups of three or more humans interact") this would be a golden opportunity to emerge from the marketing disarray and lack of clear strategy of the last two years with a turn that would leave our competitors without a card to play. I

[244] Scott Heath, the chief architect of Release 10.

would envision Malcolm Davies walking onto the stage in Chicago at the opening session of AutoCAD Expo and saying something like:

“Up to now, Autodesk have been secretive about what we were doing. No more. At this show we are introducing AutoCAD Release 10, extending AutoCAD to fully general three-dimensional model creation. We are also showing AutoCAD on the Macintosh II, OS/2 Release 1, the Sun 386i, and Sun 4, providing a wide variety of hardware options for AutoCAD users who are presently limited by existing personal computer hardware and operating systems. In the sessions that follow we will describe the directions we see for the future development of Autodesk products, and invite your suggestions and comments. From here on, we’ll talk openly with you about the composition and scheduling of Release 11, Release 12, and those that will follow. You, our customers, have defined the features we have included in AutoCAD ever since we started, and we seek a two-way dialogue with you.

“You have honoured us by choosing our products for your design work, and your choice has made AutoCAD the worldwide standard for computer aided design. Now it is time for all of us: Autodesk, the users, the resellers, the developers of applications, the manufacturers of hardware that supports AutoCAD, the user groups, the educational institutions that teach AutoCAD, and the authors and publishers of books about AutoCAD, to work together to continue to refine, expand, and develop this standard we have in common. We hope that by making our development plans public, we will stimulate the kind of two-way interchange with the rest of the AutoCAD community that will result in a better product which will serve us all well.”

So that’s the case for going public with the future of AutoCAD, and how much like “going public”[245] it is: full disclosure, high risk, and much greater visibility after you take the step. But we did pretty well by going public in the financial sense, and we might do just as well by taking that step in the product development arena as well.

If we end up adopting this path, we should immediately undertake the перестройка which will be required to roll out this policy at Expo.

[245] Making the public stock offering, with all the disclosure in the prospectus.

Where's It All Going?

One of the hardest things for people who haven't been intimately involved in the computer business for at least a decade to grasp is the nature of the exponential growth underway in our industry, and the consequences of continued exponential growth. I believe that many technology-oriented companies fail because the financial or management-oriented executives who run them don't understand how rapidly the underlying technology is developing, and how quickly their companies must change to survive in such an environment.

Every now and then it's worth reminding ourselves that we're on a wild rollercoaster and that failing to aim high enough is as sure a prescription for disaster as falling off. Here's a view of the immediate future I presented at a company meeting in early 1988.

Remarks for the March Company Meeting

by John Walker — March 4th, 1988

Recently, a lot of people have been asking me "where's it all going?". Well, I've been spending some time thinking about that, both because if you get paid to design products, it's nice to know something about the world in which they'll be used, and also because it's embarrassing to have to answer that question "Beats me".

I guess you can't live in Marin County for 14 years without being prone to having "flashes". I was driving through San Rafael last Friday, and suddenly I flashed on this.

"Hey, this *is* the future."

As a typical 1950's technoid kid, I lived for the future. You know: *Mechanix Illustrated*, *Analog* science fiction, "Science Fiction Theatre", "The Outer Limits". Now you drive through San Rafael and what do you see? On this block there's a computer store. Around the corner, a software store. Down the street, there's a hologram store. And two blocks over there's a satellite ground station store.[246]

There's moon rocks in the museums, a space station in orbit, computers on our wrists, and telephones in our shoes. So now what? Would you believe....

But first, what does this have to do with Autodesk? In thinking about starting Autodesk, I proceeded from two basic assumptions. One: it takes money to develop creative and innovative products and bring them to market. Two: it takes creative and innovative products to make money, or at least to have any fun making money. So when I thought about how to build a successful company, it seemed pretty obvious: hook up a creative engine

[246] And near the freeway, there's a store selling "clones—less than $1000". I assume they mean PC clones. I didn't go inside.

to a cash machine and throw the switch. And hey, it not only works, there's money left over. Kind of like a high-tech cat and rat farm.

So what's going to be happening to this industry we're in? Things are going to get very weird, very fast. We all know what an IBM AT is, and what it feels like to use one. The current top end machines, such as the Compaq 20/20 and the Sun 3, run about ten times as fast as an AT, and it's that kind of performance that lets us go in for solid modeling, shading, and full 3D on a desktop. But it isn't going to stop there. A couple of weeks ago I took a little machine[247] home that runs *forty* times faster than an AT.[248] And let me tell you, that feels totally different from anything you've used before.

But it isn't going to stop there.

The two magazines I always read are "Aviation Leak"[249] and this one, *Electronics*. I'd just like to quote a few items from just the last two issues of *Electronics*.

"Stellar Computer to introduce $75,000 to $100,000 machine in the next few weeks. 20–30 million instructions per second, 40 million floating point operations per second." That will run floating point code like AutoCAD about eighty times faster than an AT.

"Motorola announces a totally new RISC chip for this spring. First shipment will be 17 MIPS, but we'll show our customers how they can hit 50 this year". That's about a hundred and fifty times faster than an AT for our kind of work.

"Ardent launches first supercomputer on a desk. Runs up to 64 million floating point operations per second. Less than $150,000." Call it 200 times faster than an AT.

"Apollo unveils a desktop supercomputer. A true 64 bit workstation, it hits up to 140 million floating point operations per second. Priced below $80,000." Oh, say 450 times faster than an AT.

And where do *those* guys say they're going? After Cray, who's up there at about 3,000 times faster than an AT. And Cray's heading for 30,000, with IBM, ETA, Fujitsu, Hitachi, and NEC in hot pursuit.

And it isn't going to stop there.

Now you might ask, "What in the world do all of these $100,000 to 16 megabuck machines have to do with little ole Autodesk?" The answer lies in the fact that if you open up one of those gilded crunch-o-matics and look inside, you discover that there's really nothing in there but sand and profit.

We aren't running out of sand, and competition will take care of the profit.

So what this means is that the products that we start to develop today will, in the middle part of their life cycle, have machines available for less than ten thousand dollars, which run from one to five hundred times faster than the IBM AT—the platform that carried AutoCAD to its initial success, and from ten to fifty times faster than anything our customers have access to today.

Changes of that magnitude mean much more than nozzles appearing on the screen in 50 microseconds. They portend another qualitative shift in the kinds of products we can deliver to our customers, and those products will change the engineering design cycle at least as much as all of the computerisation to date.

---

[247] A Sun 4/260.

[248] As measured by my floating point benchmark.

[249] *Aviation Week and Space Technology*.

Let's try to look at an Autodesk customer in two or three years. This person will be using a computer that looks little different from those we use today. It'll probably have a larger screen, and that screen will have higher resolution. But look at what's on that screen! All controlled from a common user interface, into which additional products can be plugged like chips into sockets, this user may be dragging the diameter of a weight-reducing hole in a mechanical link. In one window, the shaded image is updated in close to real time. In another, the weight and centre of gravity is being recalculated as the hole changes size, and in a third window, running a little behind, a stress analysis is being run on the part with the load displayed in colour. The design of the part may be overseen by a constraint manager that digests the design rules that bound the problem, and not only helps the designer move among acceptable designs, but even offers advice when a design limit is hit. In minutes, the designer can produce a photographic quality image of the part. And naturally, the manufacturing drawings are automatically updated as the design changes.

The designer will have access to a vast database of design data, will be able to track the design history of a project, and will be able to access large bodies of data and to work with others with the computer facilitating their collaborative endeavours, thus aiding in the evolution of knowledge and the development of sound designs.[250] All of this will shorten the design cycle, increase productivity, and feed back into the process of creating the tools. And, we get *paid* for it.

Autodesk is creating incentives that spawn vibrant, burgeoning, and highly competitive markets for add-on products, extensions to our product line, and new hardware platforms. This will continue to exponentially expand the armamentarium of tools the designer can apply to create better products and bring them to market faster. Those who attempt to define a "total solution", or "control the market" will continue to discover what Autodesk has been teaching them for the last half-decade: that an open system unleashes the talents of tens of thousands of creative people each motivated by their own self-interest, and that their energy and dedication will carry that open system to successes unimagined by those who think they can "plan for the future".

Variation, selection; replication, extinction; innovation, competition; it's been working for billions of years, and it isn't going to stop tomorrow.

I think that most great business successes are the result of somebody tripping over an exponential curve, driving a spike into it, and holding on for dear life. That's what we're doing now. And there's no sign that our curve is turning back down, or that we're losing our grip. Every part of that utopian dream system is the logical outgrowth of work which is already underway at Autodesk, and by the time it comes together in, say, two years, there's no doubt in my mind that the computer power it needs will be in the hands of our users.

And the beauty of it is, we don't have to disappear into a corner and make something new. We will evolve our way there, product by product, release by release, feature by feature, with the market we've created guiding us and everybody else in it toward the best solutions to the problems it faces.

Well, time's up. Enough of the distant blue-sky horizons 24 months from now. We have to get back to the future.

[250] Such as a Xanadu system, for example. The Xanadu deal was underway but not announced at this point.

Every now and then I try to push a new feature of AutoCAD off the deep end to see if anything gives way. After adding three dimensional space polylines to the AutoCAD Release 10 development version, I used a map database to build a three dimensional model of the Earth's surface. This model, over 3.5 megabytes on disc, can be viewed from any point in space with an AutoLisp routine. This full-Earth view is centred on Cleveland, Ohio.

Xanadu

Many people within Autodesk have long been aware of Ted Nelson's work and the efforts to implement the Xanadu hypertext system. Before his involvement with Marinchip and later Autodesk, Lars Moureau met Ted Nelson in Stockholm and was instrumental in getting his books translated to Swedish and published in Sweden.

At the Hackers' Conference in October 1987 I spoke to Roger Gregory, President of Xanadu Operating Company, which was formed to implement Xanadu, about Autodesk acting as a beta site for a commercial Xanadu system. As we continued the discussion there and at several subsequent meetings, it became increasingly clear that a partnership between Xanadu and Autodesk might be beneficial to both parties; Autodesk had the financial resources and distribution to implement and launch the product and Xanadu had the technology and talented people who could build it. A Xanadu system could easily solve the data and project management tasks that Autodesk realised it needed to solve for users of its products, while offering them much more than just a drawing manager.

We eventually decided that Autodesk would purchase 80% of Xanadu and fund the development of the product. Ted Nelson came to Autodesk as a Distinguished Fellow, both to guide the development and promotion of Xanadu, and to explore other areas of research. What follows are the remarks I made at the press conference at the West Coast Computer Faire where the alliance between Autodesk and Xanadu was announced.

Statement for the Autodesk/Xanadu Press Conference

by John Walker

April 8th, 1988

The age we live in has been called many things: the Space Age, the Computer Age, the Atomic Age, the Age of the Microchip. But perhaps the most accurate appellation is "the Information Age".

Most of us have witnessed, in our own lifetimes, the unraveling of the genetic code, the uncovering of a new level of structure of matter which may, at last, prove "fundamental", cosmology transformed from a branch of philosophy into an experimental science, and through exploration by resourceful robots and brave men and women, our base of knowledge growing from one Earth to dozens of worlds.

As our fundamental knowledge has grown, so have our capacities to apply that knowledge through new technologies. It is now obvious to any thoughtful person that our powers are so great and our resources so large, that sound judgement based on accurate information is the most essential ingredient in the continuing progress of our technological civilisation. Yet often it seems that this unprecedented explosion of information has outstripped

our ability to store, process, think about, distribute, and apply it. As our future becomes more dependent on making the right choices in highly complicated problem domains, the nineteenth-century means of communication we still largely use: professional journals, libraries, and mail, become increasingly inadequate as vehicles to store, transmit, and mediate the growth of human knowledge.

Over two decades ago, the idea that there might be a solution to the problem not yet named the "information explosion" began to form in the minds of a few technologists. The confluence of technological advancements in computers and communications, coupled with an understanding of how humans distill knowledge from a sea of information and the centuries-old literary tradition led these thinkers to imagine a new medium to represent knowledge. Theodor Nelson, one of the first to explore the potential of this medium, coined a name for it, "hypertext", and gave its realisation the name "Xanadu".

In 1964, Xanadu was a dream in a single mind. In 1980, it was the shared goal of a small group of brilliant technologists. By 1989, it will be a product. And by 1995 it will begin to change the world.

Much work remains to be done to realise the potential of Xanadu—it will take the Xanadu development team 18 months to field the first Xanadu system. This will be followed by a steady stream of releases bringing added power, capacity, and flexibility to Xanadu users. As these first users explore the potential of Xanadu, they will help define the market for true hypertext systems, a market which barely exists today. But as Autodesk has demonstrated, much can happen in five years.

Five years ago Autodesk began delivering computer-aided design technology to the desktops of designers around the world. Our first product, AutoCAD, has become the de facto standard for computer aided design, and has created a large market for computer aided design tools for personal computers. Autodesk's sales have grown from $17,000 in 1982 to more than $79 million in 1987, and Autodesk has been named the Number One Hot Growth Company in America by *Business Week* magazine in each of the last two years. Autodesk's success has come from identifying new ways the personal computer could be applied to solve the problems people encounter in getting their work done, then developing, marketing, distributing, and supporting products which solve those problems.

We believed in 1982 that the time was right to deliver computer aided design to any creative individual with a personal computer. Today, we believe it's time to place the dream and potential of hypertext in the hands of the millions of people whose productivity, creativity, and achievement can be amplified by such a tool.

We believe that Xanadu is not just the first hypertext system, but the only system that has the potential to serve the individual, the workgroup or small company, the large corporation, and eventually the entire world as a repository for information in all forms: text, graphics, sound, animation, scientific and engineering data, and more. Xanadu acts as a tool for the human minds which must find order among the chaotic barrage of information that inundates us. Xanadu allows people to work together more productively in ways that existing electronic mail, databases, and online services only hint at.

Xanadu has reached the point where theory must move into practice, design into implementation, and prototype into product. Autodesk and Xanadu are forming this alliance because we share the belief that Autodesk's financial, marketing, distribution, and manufacturing resources, coupled with the vision, concrete design, implementation experience, and the talent and energy of the Xanadu implementation team can bring an initial Xanadu system to the market within 18 months.

The Xanadu design is unique in that it rejects from the outset all limits on generality, capacity, and extensibility. Implementing it in its entirety will be difficult, protracted, and expensive, but no system less ambitious can be as useful, as powerful, or as important for the long term.

Following the first Xanadu product, we will, as with AutoCAD, follow the guidance of the market in delivering increasingly powerful Xanadu systems to an ever-widening community of users. Our near-term commitment to deliver small-scale Xanadu systems in no way signals a retreat from the dream of the Xanadu global library. Autodesk and Xanadu are embarking on this venture with the goal of expediting the achievement of Theodor Nelson's original vision, and to build a successful business which will continue the development of Xanadu to the global library and beyond. As with AutoCAD, every byte of information stored in the first Xanadu system we ship will be able to move onward to future systems, new computers, and growing networks, and will take its place in the ever-growing body of human knowledge stored in the Xanadu system.

As with AutoCAD, Xanadu is an open architecture system. Our access protocols will be fully disclosed and published. We will encourage others to invent innovative ways to encode, store, retrieve, and organise information stored in our ever-growing address space. We will rely on the judgement of customers in a free market to decide what are the best approaches, and we will heed that judgement as the best guide for the future development of the Xanadu system.

The promise of hypertext is self-evident. The visionary scope and technological soundness of the Xanadu design implements that promise in its fullness. The talent of the Xanadu team and the resources of Autodesk will bring that design to market in 1989.[251] And the Xanadu era will begin.[252]

[251] Come 1992, the "resources of Autodesk" were still funding "talent of the Xanadu team" which had not, as of that date, produced anything remotely like a production prototype—in fact, nothing as impressive as the 88.1x prototype which existed before Autodesk invested in Xanadu. On August 21, 1992 Autodesk decided to pull the plug and give its interest in Xanadu back to the Xanadudes.

[252] Soon. Real soon.

Ted Nelson Joins Autodesk

This is the press release we issued on the event of Ted Nelson joining Autodesk.

Autodesk Announces New Fellowship

Sausalito, California, April 8, 1988 — Autodesk, Inc. announced today the appointment of Theodor Holm Nelson as a Distinguished Fellow. Ted will work in the Technology department at Autodesk's Sausalito office. Mr. Nelson, a world-famous visionary in computing, is one of the original developers of the hypertext concept and coined the term "hypertext" itself in the early 1960's. In 1979 Mr. Nelson joined with several others to found Xanadu Operating Company to develop hypertext as a commercial reality. Autodesk announced earlier this week that it had agreed to acquire an 80% interest in Xanadu.

Ted, a prolific author, has written numerous volumes, two of which are considered classics in the field: *Literary Machines* and *Computer Lib/Dream Machines* (recently re-issued by Microsoft Press). "Autodesk has always been committed to be the technological leader in the software industry," said Eric Lyons, Director of Technology at Autodesk. "We look forward to Ted's agile and prolific mind helping us to achieve that goal."

Ted Nelson—the man **Time** magazine called "[one of] the brightest stars in computerdom", the man Howard Rheingold, in *Tools for Thought*, called "the most outrageous and probably the funniest of the 'infonauts' ", the man *Playgirl*'s American Bachelor's Register called "the mad poet of computerdom"—Ted Nelson, maverick software designer, flamboyant legend, idealistic prognosticator, eloquent generalist, and now Distinguished Fellow at Autodesk.

Ted Nelson is where he belongs.[253]

[253] He remained an Autodesk fellow until the fall of 1992.

Bored of Directors

Having unloaded the job of President on Al Green in November of 1986, I was still left with the title of Chairman of the Board. As a consequence, I still ended up being asked to "say a few words" from time to time, to come up with comments for press releases (just try being pithy, quotable, and bland all at the same time), and I continued to be the recipient of letters and phone calls from virtually everybody who our company managed to annoy in the course of its business.

Since I am, by nature, about as gregarious as a moray eel, this began to take its toll on my ability to get productive work done. So, in early 1988, I decided to get rid of the silly Chairman title at the next annual meeting so I could program without being bothered. Since we are a public company and Everything Must Be Disclosed, we had to issue a press release bruiting this momentous announcement. Of course one must adhere to the standards of propriety on the financial wire, so we couldn't put anything interesting in the public release. That didn't stop Dan Drake from penning the following internal announcement, which was distributed within the company on April 15th, 1988.

Today or Monday we'll put out a press release, worded more or less respectably, to the same effect as what follows. What follows is the more truthful version for the benefit of Autodesk people. It speaks for itself; therefore, I'll stop speaking for it.

• • •

Autodesk, Inc. announced today that John Walker, a founder and Chairman of the Board, will not stand for re-election to its Board of Directors at its annual meeting on June 10. Alvar J. Green, the company's President and Chief Executive Officer, will assume the title of chairman. The board will be reduced from six members to five.

Mr. Walker remains with the company as a software developer, a role he has filled continuously since the company's inception in 1982.

Walker declaimed, "Ever since Autodesk was founded, my primary responsibility has been identifying product ideas, designing, implementing, and bringing products to market. This has been my most important contribution to Autodesk and it's what I do best. My goal in founding Autodesk was to build a large, profitable, financially strong, and professionally managed company that could turn product ideas into successful products that open new markets. When serving as president of Autodesk began to interfere with my ability to do these tasks, I stuck Al Green with the job. He's done it far better than I could have, and he's led Autodesk to its greatest successes. Now I find that far too many pithecanthropoids still think I 'call the shots' or want to use me as figurehead for a family of companies operating in six countries, made up of more than 400 independent, talented, and dedicated people. Consequently, far too much of the time I should be devoting to product development is being wasted

on nonproductive tasks, merely because I'm Chairman.

"Now that Autodesk has achieved market leadership with AutoCAD, it is even more important that Autodesk introduce a wide variety of new products and enhancements to existing products, broadening the market for Autodesk products. By eliminating the distractions engendered by serving as Chairman, I can devote myself full time to these clamant tasks. Believing, as I do, that our industry progresses through innovation, not litigation, I will henceforth devote all my efforts to product development."

Daniel Drake, Executive Vice President, remarked, "While he was president and chairman, John invented the AutoSketch and AutoShade products and a number of other things, some of which aren't ready to be disclosed yet. That doesn't mean he wrote a spec in his spare time for some robot to implement; he wrote the first working version of the programs and worked with other programmers through the long process of product development and release. Since that's what he likes to do best, and giving dumb speeches, reading reams of legalese, and talking to boring people on the telephone is what he likes least, this change increases the likelihood that Mr. Walker will continue to effectively contribute to Autodesk's success in the coming years."

Mr. Drake deprecated the suggestion that Mr. Walker would "do a Mitch Kapor": sell all his stock and start a new company. "Whatever you say about John, he's quite bright enough to learn from experience. Besides, I'd kill him", said Mr. Drake in a paragraph he wishes he could include in the release.

Autodesk designs, manufactures, sells, etc. all the boilerplate that we put at the end of a press release.

Valedictory

The Autodesk annual shareholders' meeting on June 10th, 1988 marked the point where I relinquished the title of chairman (or as I usually put it, always afraid of offending, "chairbeing"). Since I had held that title since the inception of the company, and since my transition to total focus on the technological future of the company was now complete, the meeting seemed an apt time to sum up the first six years and comment on the perceived opportunities which compelled me to concentrate on software development as my best contribution to Autodesk's future.

Since this will be my last meeting as chairman, I'd like to take a few minutes to share with you my view of where the company is, how we got there, where we're going, and what I'm going to be doing in the future to see that we arrive safely.

When I was thinking about starting Autodesk in 1982, the local paper[254] ran an article about software companies in Marin County. It said that the opportunity to start new ones had closed because "all the basic tools had been put on the computer".

This didn't make any sense to me. It seemed obvious that the new wave of mass-produced and mass-marketed personal computers would create ample opportunities for new products to perform tasks made possible, for the first time, by those new machines. In fact, the small group of people who founded Autodesk identified five such products, and we decided to work on all of them and see which one took off. One of our five products was AutoCAD—and its success in creating a new market for a new tool on the personal computer is what has brought us all here today.

Over the six-year history of Autodesk, technological progress in delivering affordable computer power to the mass market has, if anything, accelerated. The products we're developing today can expect, in the middle of their lifecycles, to run on computers 100 times faster than the computer that ran the first copy of AutoCAD—yet they'll cost no more than that first system. By the time Autodesk celebrates its first decade, we can expect to see affordable computers ten times faster than that entering the market, and there is no reason to believe it will stop there, either.

All of this computer power is pointless unless we can think of something to do with it. Who cares if you can recalculate your spreadsheet in 10 microseconds instead of 10 milliseconds? Fortunately, there is no shortage of tasks to which this kind of computational power can be applied—and Autodesk is superbly positioned to take advantage of this opportunity, which will create many entirely new product categories and markets, just as the IBM PC generation enabled us to create AutoCAD.

[254] The *Pacific Sun*.

Let's think for a moment about what it is that Autodesk does—at the highest level. All of our products are basically in the business of putting models of real-world objects into a computer, and then letting you do things with them. This is a fundamentally different, and more interesting, business than twiddling numbers on a spreadsheet, shuffling text in a word processor, or whatever boring things a database does. It's different because every increment of computer power, graphics performance, large low-cost storage, or new computer architecture lets us get closer to a complete model of a real world system. As we continue to approach the ideal of a complete model, we're able to do more things with the computer model, and these things turn into products we can sell, to users who already understand the need for those new tools.

AutoCAD started out as a simple two dimensional drafting system. When we got computers with more memory and faster processors like the 80286, we were able to make AutoCAD programmable. This allowed us, and hundreds of third parties, to create applications that customise AutoCAD for vertical markets. Our own AutoCAD AEC Architectural has become the leading architectural design product in the United States, and we expect our recently-introduced AutoCAD AEC Mechanical to similarly dominate the heating, ventilation, and air conditioning market.

As AutoCAD was extended into a three-dimensional modeling system, we were able to use those models to generate realistic shaded pictures. The advent of low-cost, high-resolution, colour displays enabled us to introduce AutoShade, which is placing shaded rendering in the hands of thousands of customers who previously thought it far out of their reach. And recently, we've shipped AutoFlix, which adds animation to the list of things you can do with that model inside your computer.

And still our models continue to become more accurate. AutoCAD Release 10, announced at the recent AutoCAD Expo trade show, dramatically increases AutoCAD's three-dimensional capabilities, adding surface modeling and multiple views. AutoSolid, now shipping, delivers true solid modeling to the mechanical designer. From an AutoSolid model, you can calculate mass properties, perform finite element analysis, and begin to forge the direct links from design to manufacturing which are so essential to increasing manufacturing productivity.

We also recognised that many problems don't require a tool as powerful and complicated as AutoCAD, so we introduced AutoSketch for less than $100. It can put computer aided design in the hands of anybody who draws. And we've sold more than 60,000 of them.

But you know all that... what about the future?

I believe that technological progress now underway is creating opportunities, as large or larger than the 1982 prospects for AutoCAD. Further, I believe that Autodesk is uniquely positioned to exploit these opportunities and thereby to prosper.

One of the great things about the CAD business is that you never run out of a desire for more powerful computers and more comprehensive modeling software. Extending the degree to which our products approximate reality, making models easier to build and manipulate, and enriching the ways in which one can interact with these models can easily consume all of the computer power we can envision, as far as we can foresee. This means that adding capabilities to all of our current products, adapting them to take advantage of developments in computer hardware, and extending our product line by adding additional modeling, rendering, and analysis tools will keep Autodesk's product family growing in size, and will open new markets and application areas for our products.

In addition, more powerful computers and networks will allow our customers to integrate the design, drafting, analysis, and manufacturing functions much more closely. Autodesk is the only software vendor whose products maintain complete compatibility across all computer hardware, and our commitment to open architecture makes our products the obvious choice for this integration. We are developing our products so they work equally well

for the customer with one isolated desktop computer, and the customer with 10,000 workstations distributed around the world. Both are key to our success; both are central to our strategy; both deserve the best products we can design.

Most industries are founded on a single, simple idea, such as “cheap cars for everybody” or “machines to automate business”. Autodesk remains the only software company committed to “putting the real world inside the computer”, and we believe this idea has potential as great as those that spawned the great industries of the past.

There’s a lot more to the real world than houses, turbine blades,[255] and circuit boards.

The real world is also a world of words and ideas, images and information. When we recently acquired 80% of Xanadu Operating Company, we undertook the task of putting that world into the computer, as well. The dream of hypertext, and the ability to store, organise, retrieve, annotate, and present information in ways that do not mimic paper, but are fundamentally better, has inspired a generation of thinkers. We believe that Xanadu can benefit anyone who reads, writes, and thinks, and is applicable at scales ranging from an individual’s personal computer to a global library storing all of human knowledge. We expect to ship the first commercial Xanadu system in less than 18 months. And Theodor Nelson, who invented Xanadu and coined the word “hypertext”, has joined Autodesk to help us do it.

Ideas are precious and rare. They should be immortal. Most ideas are lost, most discoveries ignored, most interrelationships unremarked upon because we have no effective way to store and then find them. Galileo observed Neptune, but nobody noticed until the 1980’s.[256] Most ideas are committed to bits of paper, obscure professional journals, or computer media made obsolete by the next wave of innovation and are lost forever. We can’t afford to lose 90% of the products of our collective minds. Xanadu is intended to fix this, and we believe that in time it will become as universal a product for people who think as television is for people who don’t. And if this happens, we will sell *lots* of them.

Autodesk’s current products as well as Xanadu are converging to address another large opportunity—building tools that help people work together more effectively. We talk about personal computers. But how many people, even free-lancers, work in total isolation? The computer will come to play a role as fundamental in communication and collaboration among people as in facilitating the work of an individual. To succeed, it must enhance individual creativity, not stifle it.

Too many computer systems seem like one-lane roads where all must proceed at the pace of the slowest. We need instead to design freeways for the mind, where people can work together, each proceeding at his own pace, without impeding or imperiling others. Products that do this effectively will yield rapid and dramatic productivity gains, and create new product categories in the software industry.

As our computer models become richer, and the ways we can manipulate them increase, creativity in devising how we interact with these models will be rewarded by making them accessible to many more people, broadening the market for our products. The AutoCAD customer who, in 1982, squinted at a low-resolution screen while typing in commands from a keyboard, can today create animated colour movies of three dimensional models built largely by pointing to a screen. I believe that progress in user interfaces has only begun.

[255] An in-joke. Shortly before I gave this speech, Autodesk had mailed out a newsletter with the Voyager spacecraft drawing (see page 354) on the cover. The picture was captioned, “Turbine blade.”

[256] Galileo’s observation of Neptune was discovered in a study of his notebooks in the early 1980’s by astronomer Charles Kowal and Galileo historian Stillman Drake—Dan Drake’s father. I also refer to this in the story *We’ll Return, After This Message*, found on page 504.

While the faltering leaders of the last technological revolution are suing each other over how their screens look, the pioneers of the next are developing tools that will take the user *through* the screen, and allow direct interaction with the data in the computer, whether as concrete as a connecting rod, or as abstract as the history of revisions to a document. Using a computer involves suspension of disbelief just as much as reading a novel. The ways people will use the tools and models we create will continue to evolve, just as styles in writing and filmmaking change with the times. Innovation spawns progress; litigation impedes it.

So what does the future look like for Autodesk? We're a software company. We build and sell tools. As long as there is demand for new tools, there will always be opportunities for great successes with software products. Anybody who thinks that all possible software tools have been invented is unimaginative, uninformed, or just stupid—man has been called the animal that creates tools—we've been at it for millions of years, and I don't think we're going to tire of it this quarter, this year, or this millennium. Autodesk has prospered so far from a tool that solves a fundamental problem—drawing and design. Autodesk will prosper in the future by continuing to develop that tool, and by creating and selling new ones that solve other problems as profound and fundamental.

The newspaper's claim in 1982 that "all the tools had been done" was actually profound as well as silly. The software that can be done at any point in time depends upon the computer power then available in the mass market. Computer power is now rising exponentially at constant price, and Autodesk is one of the few companies working seriously on products to take advantage of it.

Developing new products is difficult. I know—we've done 8 since 1982, but if we apply the same humility we had during the evolution of AutoCAD, I believe we'll continue to develop products which solve new problems and create new markets for the personal computer. In 1983 people used to ask us about "our vision of the future of CAD". We used to say in all candour, "we don't really know enough about CAD to have one". But we had a secret. We knew somebody who *did* know—our customers. No vendor in any market knows as much about his product, how it is being applied, where its shortcomings are, and what is needed to make it better, as the customers using it. And today, we have more CAD customers than any other company in the business, and they continue to guide the development of AutoCAD and all our other products. Hubris, borne of detachment from the real problems facing customers, has doomed many technology companies. That's not going to happen at Autodesk, because we still don't know enough to chart the future of anything. But we listen patiently, and we take lots of notes.

So we don't know what the future will bring, but we do think we know how to get there. This company was founded on the idea of developing 5 products and getting behind whichever one took off. I like to be able to be wrong 80% of the time and still do well. It worked, and we will continue to follow that evolutionary approach to the marketplace as we have for six years. Many things we try may not work, but we expect those that do to form the foundation of our future prosperity. This isn't all that unusual an approach—every investor with a diversified portfolio is doing the same thing. I think that companies don't like to admit they're being guided by the market because you get credit for being a brilliant planner, not for being a careful observer. Evolution looks like planning when you look back in time—that's why so many people have trouble understanding it. In fact, market-driven pragmatism not only works; it *pays* well.

Getting there won't be easy. But getting *here* wasn't easy. We've seen competitive products from IBM, CalComp, AT&T, and Computervision come and go. We've lived through the worst stock market crash in history. We've seen overt theft of our products, and we've discovered how hard it is to stop it. We've made misjudgements of the market and we've corrected our course. Sometimes we've been wrong, but we never stayed wrong too long. I'm not going to stand here and tell you that Autodesk will never have a bad quarter or

never screw up.[257] But I do believe that as long as Autodesk remains focused on the enormous opportunities we face, as long as we act prudently and responsibly to take advantage of them, what will be called Autodesk's period of great growth and success lies in the future, not in the past six years. And it is a measure of my confidence that the management and directors of Autodesk will achieve this that I have no hesitation whatsoever in entrusting the operation of our company to them.

Many of the things we need to do are obvious. But just because they're obvious doesn't mean they're easy. It is because I believe that the technical work ahead of us is so great, so important, and the benefits to Autodesk are so enormous, that I will be focusing all of my energies on these tasks, swearing off meetings with lawyers, giving speeches, signing checks, filling out forms, talking to analysts, and all of the other fun things that I got to do while chairman.

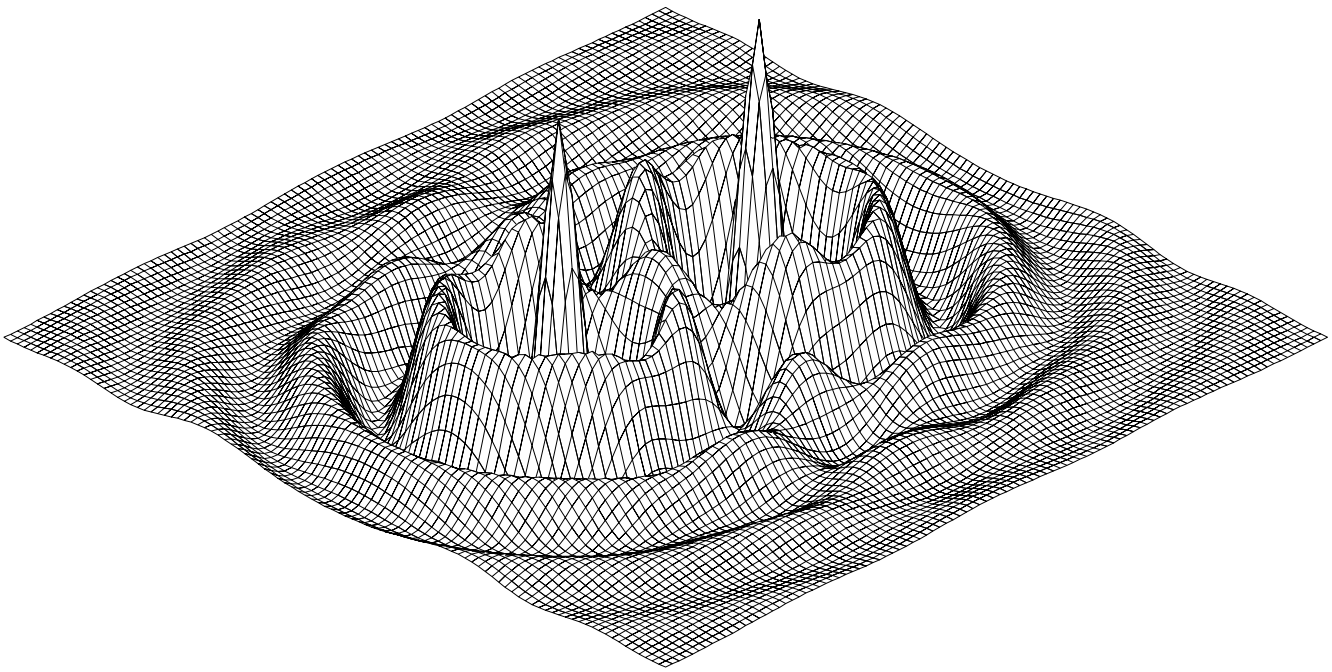
For whatever reasons, we've all wound up owning a company that's right in the middle of an exponentially growing technological revolution that's remaking our world. Many companies share this position with us. But we know where we are—and we think we know what to do about it.

Many companies have abandoned an optimistic view of the future. Starting a company out of thin air at the bottom of a recession with almost no money and seeing it grow into this in six years, largely as a result of simply plugging away day and night on the mundane, near-term tasks that had to be done, is a wonderful antidote for pessimism. It gives you great confidence in continuing to reach for further success and achievement that same way—getting all the details right, but never losing sight of where success can carry us. Without imagination we are all doomed to live in a grey world devoid of hope and excitement, fortune and glory. We talk about the Golden Age of Engineering. It's our age—if we work to build it. To achieve it, we must toil tirelessly at the gory details. If we invest that effort, our achievements will be unbounded, our horizons endless, and the potential for Autodesk incalculable.

Thank you for the confidence and trust you've placed in me the past six years.

[257] Indeed.... See page 718.

The ongoing development of AutoCAD is extending it into a general three-dimensional surface modeler. AutoCAD Release 10 adds polygon meshes, providing a general surface object. This allows new applications, such as modeling the surface defined by functions of two variables. Here's an interference pattern generated by two exponentially damped cosine waves. I made this with a little AutoLisp program to test huge polygon meshes with interesting cases.



L'Envoi: the First Six Years

There you have it: six years of Autodesk. While the record is far from complete and, like all collections of raw sources, lacks the balance and perspective of a historical narrative, reading these documents can't help but recreate the excitement, the fear, the exhaustion, the tension, the uncertainty, the exhilaration, and the feeling of accomplishment that we all shared as we lived through the times that these papers bear witness to.

I certainly know that collecting and editing them reminded me of many things I had forgotten. It's always easy to believe that things were much easier in the good old days, or to wish that we could get things done as quickly and easily as we used to. These papers show that at every point in the company's evolution we faced difficult decisions whose effects could not be predicted but which had to be made immediately based on incomplete information. They show our constant struggle to be responsive to our customers, dealers, and stockholders. They show that the process of getting high-quality products into the hands of users in a timely manner can never be "easy". They prove beyond any possible dispute that the one asset that is responsible for the success of this company is the people in it: people who have always been willing to exert whatever effort was required to get the job done and to do that job to the best of their ability because they believed in what the company was doing and they believed they would be rewarded for their exertions.

Sometime in 1984, when the company had begun to make a mark in the industry, a reporter asked me if the company had a "philosophy". I hadn't really though about it in those terms—it seemed to me that we were just doing what made sense. Like all reporters, this one pushed me for a pithy quote, so I said:

> Make the best product.
> No bullshit.
> Reward the people that do the work.

A little more than six years ago Autodesk was all potential and no success. As the years have passed our efforts have brought us enormous success and have vaulted Autodesk into the first rank of small high-technology growth companies in the world. This success is both the well-deserved reward for what we've accomplished so far and the springboard that is creating new opportunities of unprecedented scale today.

It's easy to kick back and concentrate on meeting the sales and earnings projections on a quarter to quarter basis and to view our existing products as the centre of our business for all time, to be incrementally enhanced over the years. And we must do these things. But to remain true to the strategy that has brought us so far so fast we must also be constantly on the look-out for the next AutoCAD: the product that comes from nowhere, in an industry that doesn't exist yet, that all of the well-respected analysts say "can't be done", or "can't be done on a PC", or "won't sell". All of these things were said of AutoCAD.

In business you have to constantly try to expand the scope of your operations and move into new areas of

opportunity. If you don't, your competitors will and you'll find yourself in a darkening corner of a market growing cold with obsolescence. If you run your business well and build the mainstream revenue sources while exploring the opportunities of the future, you can stay on the path to growth and success that dwarfs what we've achieved so far.

There was a day when General Electric, IBM, AT&T, Ford Motor Company—all of the Titans of industry, were the same size that Autodesk is now. Most of those companies took far longer to get to that point than we have, and few were in as strong a competitive and financial position when they got there. Like those companies, we're riding a technological wave which has been building for decades and whose limits cannot even be calculated today. If we continue to demonstrate the kind of creativity, productivity, and energy that we've shown so far, we can build Autodesk into a peer of these great industries. To take advantage of this opportunity we share will take vision, leadership, sound management, technological creativity, financial strength, a commitment to excellence in everything we do, and most importantly, the willingness to do the tedious work that turns opportunity into success.

Perhaps the greatest risk that faces Autodesk today is the tendency to think that our success to date is enough, or that now that we're a "large, established company" we can't afford the kinds of wild technological gambles we made in the early days. But it's those very gambles that will carry us from where we are today to the next plateau of success—and the next and the one after that. And there's one thing no reasonable person can doubt after reading this history: that the people who are Autodesk have what it takes to make it happen.

As Arthur C. Clarke said in 1963,[258]

> "Despite the perils and problems we face, we should be glad we are here at this time. Every venture is like a surf rider, carried forward on the crest of a wave. The wave bearing us has scarcely started its run; those who thought it was already slackening spoke decades too soon. We are poised now, in the precarious but exhilarating balance that is the essence of real living, the antithesis of mere existence. Behind us lie the reefs we have already passed; beneath us the great wave, barely flecked with foam, humps its back still higher from the sea.
>
> "And ahead?
>
> "We cannot tell; we are too far out to see the unknown land. It is enough to ride the wave."

[258] In *Profiles of the Future* by Arthur C. Clarke, Harper & Row 1963 / Bantam 1964, 1967.

Technological Leadership

After I escaped from day-to-day management, I had time to think about Autodesk's strategy and direction. What I found was disquieting. In August of 1988, I wrote this memo to Al Green to attempt to define the issues on the technological side of the company which needed addressing. The roots of most of the problems which were eventually brought to a much wider audience two and a half years later in Information Letter 14 (see page 600) can be seen here in embryonic form. Those who believe me a renegade leader of a "cabal" or "theocracy of hackers" may find the analysis and recommendations herein rather surprising.

To: Al Green
From: John Walker
Subject: Technological Leadership
Date: August 23rd, 1988

No person who has lived through Autodesk's growth over the last six years, or even a significant part of it, could cling to the fantasy that the process has been a well-planned, steady march to success, regardless of how much our financial results may support that view. Instead, the company has encountered and survived almost all of the crises that befall rapidly growing companies in turbulent markets and has emerged stronger and better positioned for having done so. This process of maturation and tempering can be viewed, however, as lurching from one potential disaster to another, hoping to fix each problem before catastrophe befalls our venture—and that is precisely what we've done, always managing to recognise each crisis in time and act decisively to resolve it.

I am writing this memo because I believe that Autodesk is presently facing one of the most serious crises in its history, a crisis of technological leadership calling for changes in the technical structure, direction, management, and leadership which must occur if Autodesk is to continue its success in the marketplace. This crisis is particularly acute because its resolution will involve restructuring an area of the company which has to date been the least affected by the imposition of structure and professional management as the company has grown, and in many ways is still run as it was in 1983. Consequently, imposing changes in this area inevitably will affect the relationship of founders to the company, the software development group's perception of its role in the company, and the very process by which Autodesk products are conceived, implemented, enhanced, and supported. These changes cannot be made without creating stress and some disruption, but I believe that deferring them will place at risk all of what we have worked for to date, and the great potential which I continue to believe this company possesses for the future.

I believe that Autodesk has long outgrown the kind of passive, caretaker management which has characterised software development since the company's inception. I believe that there is an almost total lack of technological direction to the company's software development efforts, that no overall design is being applied to achieve our

publicly-stated goals of product line integration, that resources are being misapplied to company priorities, that a pattern of abandonment of products once launched[259] is contributing to product line stagnation and exposing Autodesk to serious competitive threats, and that all of these are symptoms of inadequate technological vision, leadership to implement that vision, and competent professional management to accomplish the tasks required to reach our common goals. Addressing these problems before they cause even more serious damage must be one of the company's highest priorities.

In what follows I will survey the state of the company's product line, focusing on how product design, development, and support are managed. Most of the comments regarding market position and competition, and all of the sales results, are drawn from the August 1988 Marketing and Sales Plan.

AutoCAD Release 10

AutoCAD Release 10 was conceived as the generalisation of AutoCAD to three dimensions. In early 1987 Scott Heath suggested focusing our previously ill-defined and stagnant 3D project on the goal of first generalising AutoCAD to operations in any plane in space, then extending that base into a modest 3D feature set. A programmers' meeting was held on March 8, 1987 to attempt to make both the short- and long-term goals concrete, and development began shortly thereafter (with significant disruption and diversion from Release 9 final development and debugging which was underway concurrently).

During the development of the product, several major new features were developed and integrated into the product, including dialogue box versions of viewing commands, entity database handles, dynamic 3D viewing operations, multiple on-screen windows, three-dimensional polylines and surface meshes, and an extended memory version of AutoLisp. These features, and several others, were added to the product based on perceptions of competitive products, comments from potential users shown prereleases of the product, general feel for the market, or just at the urging of the developer of the code—in short, in the same way the design of AutoCAD has proceeded since Version 1.0.

Based on a statement that all coding for Release 10 would be completed by April 30th, 1988, a tentative shipment date of June 25th, 1988 was set for the product and announcement was slated for AutoCAD Expo 1988. As the announcement neared, it became clear that major code submissions were still underway—in no sense was the product development close to complete by the originally-scheduled "code cutoff", and the originally-slated release date was unachievable. As testing of the product progressed, it became evident that several serious upward compatibility issues had been inadequately addressed, and that ignoring these problems in the final product would disrupt existing applications and user-customisation, and as a consequence major effort was applied to implementing **`FLATLAND`** mode which attempts to preserve compatibility with Release 9.

A period of reevaluation and slippage ensued, resulting in a current best-case shipment date more than three months after the original date, with some programmers remaining very skeptical about the stability of any product shipped on that date. Now even if the product slips to November,[260] it will not break any Autodesk record for late shipment of a product—Version 2.6 was announced in November 1986 with an internal shipment target of January 1987 and finally shipped in April of 1987. But in the intervening time the stakes have risen enormously, and Release 10 is not like Version 2.6. Release 10 is arguably the most important release of AutoCAD ever; for years we have been assailed for our lack of 3D, and Release 10 constitutes our statement

[259] This directly foreshadows the detailed analysis "Benign neglect: silence in the marketplace" in Information Letter 14 (see page 617).

[260] Release 10 finally shipped in October 1988.

of just what a "full 3D AutoCAD" means. Version 2.6 was conceived, presented, and interpreted by the market as a minor update, so it placed much less of our reputation on the line, and the potential for order deferral if it were late was much less.

With Release 10, we are going forth to do battle with Cadkey, which we have allowed to establish itself as a strong competitor by our lack of 3D capability, and with the mainframe systems coming down, on their own three dimensional home ground. If Release 10 fails to deliver on our promises for it, is seen as slipping from release date to release date, or develops a reputation for unreliability either through user-encountered problems, application incompatibility, or the perception created by product recalls even if prompted by minor problems, it will be a serious blow to our efforts to regain the momentum in the market we have lost during the years our development efforts have ignored 3D features.

In addition, the financial stakes for late shipment and a possible recall of Release 10 are enormous—we have publicly announced the product, promoted it as one of the most significant enhancements of AutoCAD ever, positioned it as part of our major account strategy, and committed to free updates for all AutoCADs sold after Expo, creating both an update liability that grows every month and guaranteeing that any product recall, if one occurs, will be highly visible both to our users and the financial community.

What can we conclude from examining AutoCAD Release 10? The development of Release 10 to date is not a story of mismangement, incompetence, or inattention to detail. It is simply the result of Autodesk conducting its product definition, development, and refinement in precisely the same way we did for Releases 2.0 through Version 9, with similar consequences for product focus, timeliness, and quality risks.

The definition of the features in the product was largely left to the programmers writing the code. Major features were included only because individuals ran off, developed them, and them dumped them into the product stream. Features were coded, in some cases involving significant expenditures of time, then discarded when perceived as unworkable (**DDVIEW** is one example). Product definition was driven little, if at all, by formal guidance by the marketing and sales organisation, increasing the influence of random encounters with users, developers, and sales people on individual programmers. As an example of this process, Coons patches were included in the product because I had read about them in a book and wanted some kind of easy-to-create sculptured surface to demonstrate 3D surface patches. Making the data base recoverable after a random clobber, considered a major shortcoming of the product, is a project of the same magnitude and was not done because I did not think of it at the time. The following is an incomplete list of features which could have been included in Release 10 by trading off development time with some other features or by assigning additional manpower to the project but which were omitted largely because nobody argued for them. *Every one* of these features has also been omitted from a release prior to Release 10 by being forgotten until it was too late.

- Database checksums, recovery from bad databases.[261]
- Ordinate point dimensioning.[262]
- Redefine polyline beveling, including user control with round end option.
- Add triangulatisation of nonplanar and concave faces in **HIDE**.
- User-defined status line contents, just like the one that missed the 1.4 code cutoff.[263]

[261] Finally done in Release 11.
[262] Done in Release 11.
[263] Done in Release 12.

- Something for US version to prevent export, just like the DES code that missed the 2.6 cutoff.[264]
- Fast detection and elimination of duplicate entities in EID selection set composition.
- Abbreviated command names.[265]
- User-definable dialogue boxes for AutoLISP.[266]
- General entity-edit dialogue box, with provisions for editing text.[267]
- Protected layers (can **OSNAP**, and they display, but you can't select entities on them).[268]
- Text with all 9 Intergraph alignments (both edges and centre in both axes).[269]

This is not to imply that any of these items are more important than those included in Release 10, but simply to point out that they represent the "Release 10 that never was" purely by having eluded the attention of the programmers while coding, not through any conscious market-driven decision on what capabilities we need in our product to defend and expand our market share.[270]

If we seriously intend to compete with Intergraph, IBM, and Prime, I do not believe we can afford to continue to develop our products in this manner.

AutoCAD For The Macintosh

Based on our evaluation of the Macintosh II, ongoing communications with Apple, the emergence of competitive products on the Macintosh, pressure from dealers, and the intention of major accounts to install that machine, in late 1987 Autodesk committed to developing a CAD product for the Macintosh, and one of our most senior developers was assigned exclusively to the project.

Based on technical considerations which made it appear impossible to directly port AutoCAD to the Macintosh, and also from an understanding that products from other platforms merely ported to the Macintosh were unsalable, Autodesk committed to a multi-man-year development effort with the goal of implementing a new CAD product for the Macintosh. The initial operating capability of the product was anticipated to be at least a year from the inception of the project, and to be a two-dimensional product with the functionality of a much-extended AutoSketch. Feedback from Apple was contradictory; on the one hand they "needed AutoCAD to sell the engineering market" but also wanted a Macintosh-specific product which provided unique value-added based on the application commonality of the Macintosh.

I could not personally believe that Autodesk had so lightly committed to developing a totally new CAD product which, to justify the resources which would be expended in developing it, would eventually spawn marketing,

[264]This was an attempt to reduce the number of non-hardware-locked domestic U.S. AutoCADs being exported into markets where piracy was rampant. The idea was that we included something like DES encryption of the database as an option, it would be, potentially, a criminal act to export AutoCAD from the U.S., and this might act as a deterrent. In all, probably a dumb idea.

[265]Done in Release 11.

[266]Done in Release 12.

[267]DDEDIT for text and attributes was implemented in Release 11. More comprehensive editing was added in Release 12.

[268]Done in Release 12.

[269]Done in Release 11.

[270]Indeed, many customers reacted negatively to Release 10 when it finally shipped, perceiving it as containing primarily 3D features which they, as drafters, would never use. Had Release 10 contained the above features, which ended up slipping to subsequent releases, it would, in all likelihood, have been much better received.

training, support, and development costs comparable to those for AutoCAD, to address a single market whose total size, using the most generous measure, was 10% of the size of the market for AutoCAD. I also believed that in the one year estimated time to market (which I considered extremely optimistic), the window of opportunity to position a product on the Macintosh would close, particularly as Versacad was then beginning deliveries of a product with capabilities similar to those of the product we contemplated and we knew of several other entrants which would reach the market before we could.

In February of 1988, I started to investigate whether it would be possible, by various means, to port AutoCAD to the Macintosh II. By February 15th, 1988 I had demonstrated AutoCAD Release 10 running on the Macintosh II in 5 megabytes of memory.[271] Immediately this was demonstrated, the previously-committed project to develop a new CAD system for the Macintosh evaporated and was replaced by a project to port AutoCAD Release 10 to the Macintosh, while adding some Macintosh-specific capabilities. While I was happy to see Autodesk adopt what I personally believed not only a more realistic but also a more beneficial goal for a Macintosh product, the whole experience of the definition, destruction, and redirection of the Macintosh product does not show Autodesk's formulation of strategy and technical decision making in its best light.

To wit, *what if I had done what I was supposed to be doing in the first two weeks of February, rather than fooling around with a Macintosh?* Or, *what if I had not been given a Macintosh II to play around with?* Where was the serious evaluation of our alternatives with regard to the Macintosh before committing to a multi-year development effort? Where were the trade-offs weighed among that project and numerous other claims on our development resources? Why, if it wasn't a good idea to port AutoCAD when it wasn't thought possible, did it become a good idea when I proved it was possible after all?

AutoCAD Release 11

As of this date, I do not believe that anybody within Autodesk has a firm grasp on the constitution of the next release of AutoCAD. The only firm statement is that it will not run in 640K, though what it will run under on 16 bit platforms seems presently up in the air.[272] One long meeting was held which resulted in an enormous wish-list of disparate features, but little or no winnowing or aggregation of that list into concrete implementation proposals has been done to my knowledge.

To date, no implementation of Release 11 is underway, scheduled, or contemplated. I think we've grown to the point that we can't afford to develop our mainstream product in the kind of cycle that we did in the past. Typically development of an AutoCAD release has focused around one or a small group of central features. Development work would start on these features, with a small number of senior developers working on them. As development progressed, other features would be contributed by those developers or others, often chosen by shooting targets of opportunity off the wish list while making other changes in related areas of the program.

[271] I pulled this off with one of the dirtiest technological tricks in my inky-black hacking career. There was no compiler for the Macintosh which generated 32 bit code, permitted data segments larger than 32K, or compiled in-line floating point instructions: all prerequisites for AutoCAD. But, I noted, at the very same time we were shipping AutoCAD on the Sun 3, which used the very same Motorola 68020 microprocessor as the Macintosh II. So, I simply compiled all of AutoCAD except for the display, keyboard, mouse, and file I/O drivers on the Sun, linking them into one huge relocatable file which I called "The Titanic". Then, using LightSpeed (later Think) C, I wrote a small conventional Macintosh application which obtained a large chunk of memory, loaded the Titanic into this block of memory, dynamically relocating it and linking external references in it to "stubroutines" defined in the Macintosh program, and then launching it. The Macintosh program, "Tugboat," thus nudged the Titanic into place on the Macintosh. Believe it or not, this all worked (though debugging it with a joke-debugger called TMON which couldn't disassemble the 68020 instructions generated by the Sun compiler is a memory still powerful enough to make me grind my teeth), and this is how we finally (after a lot more work by many other people) shipped Release 10 for the Macintosh.

[272] Release 11 finally *did* run in 640K and was launched (without notable success) as R11 286.

As a release date came into focus, usually aimed at a major trade show, Duff would focus on documentation while other developers would devote their efforts almost exclusively to testing and fixing bugs generated by the in-house QA and beta testing process. The effort would reach a crescendo at the release date, with a coda devoted to getting all the other hardware platforms shipped. A period of exhaustion would ensue, followed by gearing-up for the next release. This cycle describes the development of Release 10 to date just as well as it does Version 1.4.

This is how most small companies develop software, and it permits efficient utilisation of limited manpower, minimisation of management overhead, and eases the always-difficult task of product configuration management. The question is whether a company in our position, facing the competition we do, can afford to develop our product in a manner that spends less than half the available time actually enhancing the capabilities of our product. I am sure that Cadkey and Versacad do their development the same way we do, but I seriously doubt that our mainframe colleagues at IBM, Intergraph, and Prime engage in this kind of half-wave development (on the other hand, maybe I'm wrong, but in any case we're the ones playing catch-up, and we can ill-afford to waste *any* time).

We now have more than 50 people in our software development department, not counting those in other departments who are dedicated to the process of getting new releases of AutoCAD out the door. Should we not move to concurrent development of AutoCAD while stabilising a release for shipment? Yes, this makes source code control, project management, resource allocation, personnel management, and everything else far more difficult. *But doesn't maintaining the technological leadership of a product that's bringing in 100 million dollars a year justify doing some difficult things?*[273]

AutoCAD AEC

This product line, which we acquired from Archsoft and have been steadily expanding our commitment to ever since, including acquiring a second product from them and then reimplementing it as well, is now seen, despite market share leadership, as falling behind the market in both functionality and performance terms. Why should it? This is a product which has generated revenue in excess of $100,000 per month for each of the last 18 months, and has recently contributed sales in the $200,000 per month range. We have an entire sub-department dedicated to development and maintenance of this product line with at least five people exclusively assigned to this product. This development team exceeds in size that devoted to AutoCAD during the period when its sales were comparable, a time during which AutoCAD clearly established itself as the technology leader in its market.

If the people working on the AEC product line are not the right people, why have they not been replaced with the right people? If AutoCAD AEC cannot deliver better performance without modifications to AutoCAD, why have these modifications not been requested, designed, and implemented? If we have achieved market leadership with this product but now believe it impossible to maintain, why are setting the same group that turned in that performance to the task of creating an "infrastructure" product which we now believe central to competing with Intergraph in the AEC market?[274]

[273] Compare this with the description of the AutoCAD development process in Information Letter 14 on page 616.

[274] In May of 1990, Autodesk reverted AutoCAD AEC Architectural and Mechanical to ASG, the successor company to Archsoft, and thereby left the architectural application business.

AutoSketch

AutoSketch was designed to give Autodesk an entry in the low-cost drawing market, both in the hope of its becoming a successful product in its own right, and to protect against erosion of the low end of the AutoCAD market by competitors who could expand from such a foothold by upgrading their products (as Generic CADD has done, precisely as we predicted and feared). The product was previewed at the first AutoCAD Expo in 1986 and shipped in November of that year. Development of the product through shipment consumed approximately one man-year, involving senior development staff exclusively. Following its shipment, the product was essentially ignored by Autodesk's marketing and sales effort as well as the development group until the establishment of the AutoSketch #1 Team in 1988. Recently some technical resources have been expended on the product, initially to create versions for hardware bundling deals and the PS/2, and now to add some user-requested features to the product.

AutoSketch is evaluated in the Marketing/Sales plan as "falling behind the fast moving market of Low-End CAD and Graphics software". No wonder—it's remarkable that a product on which no serious work has been done in two years is a contender at all; the fact that we've managed to sell 70,000 units of a static product is truly remarkable and a credit to those who made it happen. The reason given for the abandonment of AutoSketch by the development group is "Technical resources". I disagree—its abandonment is the consequence of priorities being placed elsewhere. AutoSketch is a simple program to learn and work on—I find it hardly credible that if development of AutoSketch had been set as a clear priority for the technical department and the management of that department had acted to obtain and assign resources to the product, twenty-four months would have passed with no action—months during which numerous development personnel were recruited and put to work on AutoCAD, with its much larger and steeper learning curve. AutoSketch has generated revenues on the order of $50,000 per month throughout FY 1988 and 1989 to date, with several peaks near $100,000. Yet AutoSketch has been assigned little or no development resources while AEC, with sales only twice that, has merited an entire subdepartment, dedicated promotion, etc. This is not a resource constraint; it is lack of priority, focus, and follow-through.

The product strategy for AutoSketch includes a plan to include the "Rube" constraint manager in the product to provide a unique point of distinction in the market, make it attractive to a new set of users (including those with AutoCAD), and possibly move the product up-market in price and perception. Yet the technology group who prototyped Rube ceased work on integrating it into AutoSketch over a month ago because they were unable to obtain any direction, commitment of resources, or plan of integration from the software development group.[275]

AutoShade

AutoShade, which started out as a simple interface to rendering software to be provided by Visual Engineering, and ended as an in-house product which consumed on the order of a man-year of senior development manpower, was shipped in September 1987 to provide a shaded rendering capability to AutoCAD users. At the time of its introduction, it was the first rendering product for AutoCAD models, and having sold more than 5,000 units to date, it is clearly the market leader.

After the shipment of the product, it was abandoned by the software development department. Until recently,

[275] Had this project been pursued, Autodesk would have been able to launch, in early 1989, a 2D parametric modeling package which ran on a minimal DOS configuration and served as a conceptual modeling front-end to AutoCAD. This opportunity was passed by. Two years later, in 1990, it was suggested that Autodesk should acquire Ashlar to obtain constraint technology similar to that in Rube. At this writing, in 1993, Autodesk does not have a constraint manager in its product portfolio.

when one developer on his own initiative ceased work on AutoCAD to enhance AutoShade, essentially no resources were assigned to enhance the product to capitalise on its head start and strong sales, which have ranged from $30,000 to $90,000 per month since introduction of the product. All of the weaknesses cited in the Marketing/Sales Plan except for its dependence of 3D in AutoCAD, an inherent weakness of any shaded rendering product, and the fact that it is separate from AutoCAD (an opportunity for us in the future which is not available to any competitor), are purely consequences of the abandonment of the ongoing development of the product.

User Interfaces

In a speech at the InfoCorp Silverado conference in March 1986, I said that "user interface is a *distribution* issue".[276] In other words, the economic viability of a low-cost product sold through retail channels without extensive support depends on that product being accessible to its potential customers without extensive training or handholding. One reason that AutoCAD was able to fend off assaults from PC versions of mainframe CAD software is that users were able to draw with AutoCAD without spending the time a mainframe CAD operator would invest to learn his system.

AutoCAD's user interface wasn't all that great in 1986, but it was better than most of the competition. Since that time, we have gotten much worse while our competition has been improving rapidly. Today I believe that no commercial software product with comparable installed base and revenues has a user interface remotely as bad as AutoCAD's, that the situation is degrading rapidly and spreading to our other products, and that the wages of our inattention to what should be at the heart of the design of any interactive tool are now coming due in the form of lengthy test time, unreliable applications, expensive training requirements, and rising product support costs.

Lack of vision, the absence of clear design, inattention to developments in competitive products, failure to apply reasonable engineering judgement to counter the demands of a QA department run amok in fields of casuistry, and general abandonment of the user interface to whatever seems expedient to developers as code is written, has resulted in an almost inconceivably wordy, obscure, error-prone, and virtually impossible-to-document input language which makes the CADDS 4 commands we used to ridicule elegant by comparison. It is ironic that a company whose staff ridicules products like TEX and EMACS as "suitable only for programmers" is engaged in selling an application whose command language makes either seem a paragon of clarity and generality.

While AutoCAD's command language was an object of neglect, computer vendors were rapidly raising the stakes by attempting to make the user interface part of their operating systems. It appears increasingly probable that, in order to compete in the future on multiple hardware platforms, an application like AutoCAD must become able to conform to multiple different user interface mechanisms while somehow maintaining application and database compatibility across platforms. There has been general acknowledgement for over a year of the seriousness of this problem and how important finding a solution is to Autodesk's future. During that time, no resources have been allocated to solving it within the software development department—the problem has been left to me to solve in whatever time I make available to work on it, with whatever energy I can spare from other tasks.

Our development and QA staff seems increasingly consumed in debates over arcana about which nobody outside Autodesk seems to care as shipment deadlines slide past, while our competitors are creating and delivering products which are making intuitive concepts that Autodesk seems to lose grasp of in a lake of wordy commands, and a sea of error messages whose aggregate size exceeds the total size of some Macintosh

[276] See page 302

applications. If you don't know what I'm talking about, or think this is some of the hyperbole often attributed to me, I suggest you schedule, back to back, a two hour demo in which a robot arm is constructed with AutoCAD, and a 15 minute demo in which the same task is done with Swivel 3D, a new modeling product for the Macintosh. Swivel 3D was originally developed to explore solid model interaction with the VPL glove, a user interface modality which seems to elicit laughter from the majority of Autodesk personnel, notwithstanding its being considered one of the most promising approaches to 3D manipulation by the user interface group at NASA Ames and having been the subject of the cover of *Scientific American* in October 1987.[277].

Summary, Conclusions, and Recommendations

This has not been a pleasant paper to write. Recounting these observations has opened a large collection of incompletely-healed wounds; anticipating the reaction of those who read these opinions is unpleasant to one who values placidity and amity as much as I do; and making the recommendations that follow mean in large part consigning many of the idealistic goals I had for software development when we organised this company to the dumpster.

But failing to come to terms with what I believe are the facts, and choosing not to act to remedy a situation I believe already dire and rapidly moving beyond hope of salvation, will sacrifice much more than some six year old hopes and dreams and cause far more anguish than some lost friendships and working relationships. If we allow AutoCAD to lose its technological leadership, then I believe it is inevitable, sooner or later, that its market leadership will also be forfeit. If we allow competitors to count on our abandonment of every new product entry shortly after shipment, we are expending our development resources merely to produce prototypes for our competitors to use against us. If our product development does not meet the standards expected from an industry leader, in volume, scope, imagination, quality, and timeliness, then any attempts we make to compete with established industry leaders on their own turf, workstations and major accounts, are foredoomed.

If we do not act rapidly to totally professionalise the technological sector of our business, from senior management, to product definition, to project management, to quality, then we put at risk all we have worked for so long and hard. For years we have been furiously working to catch up with the big guys, and our limited resources have made many of our decisions for us. This allowed us to adopt and survive with a passive, reactive style of management of our development department. This mode of operation is not only inappropriate in a company with a development staff of 50 people and a bank balance of $100 million, it is a prescription for disaster when our company attempts, as we must, to pull out from the pack and extend our leadership position in PC CAD to the CAD market as a whole. We must make these changes even as we acknowledge that they will make the software development department less attractive to entrepreneurially-oriented self-starters, including myself.

The issue here is not a technical department issue; it is the future of our company. The evidence that something is seriously wrong in the software development side of our business has been apparent to me for months, and increasingly people have been coming to me without prompting to share their worries on the subject. This is a problem which imperils our company at least as much as earlier crises in marketing and sales, manufacturing, and accounting. We met those crises head-on, fixed them, and moved on to greater successes. We must not shrink from this one.

[277] I addressed this in more detail and urged Autodesk to pioneer 3D user interfaces in September 1988 in "Through the Looking Glass" (see page 439)

Through the Looking Glass

I had been interested in three dimensional user interfaces ever since I first heard of Ivan Sutherland's pioneering work in the late 1960's. NASA Ames had demonstrated a modern system with head and hand tracking, and it was clearly only a matter of time until the computing power of inexpensive personal computer made virtual reality a market reality. It also seemed obvious that a company experienced in three-dimensional geometric modeling and high-performance graphics on cheap hardware was uniquely qualified to become the leader in this emerging market. In this paper I urged Autodesk to enter the nascent field of virtual reality and set the standards. The paper was well received, and a project was launched shortly thereafter. Unfortunately, the disconnection between R&D and marketing, and the general paralysis that gripped Autodesk during the late 1980's (see Information Letter 14 on page 600) caused Autodesk to squander a once-in-a-decade opportunity. Autodesk did finally manage to field a virtual reality developer kit in early 1993, but the chance to "own the market" had been lost long beforehand. An abridged version of this paper appeared in the book "The Art of Human-Computer Interface Design," Brenda Laurel, ed., Addison-Wesley, 1990 (ISBN 0-201-51797-3).

Through the Looking Glass

Beyond "User Interfaces"

Today's fascination with "user interfaces" is an artifact of how we currently operate computers—with screens, keyboards, and pointing devices, just as job control languages grew from punched card batch systems. Near term technological developments promise to replace user interfaces with something very different.

by

John Walker
September 1, 1988

A generation in computing is often identified by the technology used to fabricate computers of that era. By this reckoning, a list like the following is usually presented:

Generation	Fabrication technology
First	Vacuum tubes
Second	Transistors
Third	SSI and MSI Integrated circuits
Fourth	LSI—CPU on a chip

While fabrication technology has been the dominant influence on the cost and performance of computers, and hence has been the economic determinant of how widely they are available and to what purposes they can be applied, from the user's perspective it is virtually irrelevant. A user of a Unix system, for example, may hardly be aware of whether that system is running on a PDP–1 (second generation), VAX (third generation), or 68020 (fourth generation), and other than issues of performance and cost, could care less.

User interaction generations

From the user's standpoint, *how* he interacts with the computer is an issue surpassingly more important than *what* the computer is built from. The very way in which the user perceives the computer (his mental model of it), the degree to which specialised knowledge and extensive training are required to use it, and therefore the extent to which the computer becomes accessible to a very broad segment of the populace is largely controlled by the means through which the user operates the computer. Let's try to redefine computer generations in terms of modalities of operation.

Generation	Means of operation
First	Plugboards, dedicated setup
Second	Punched card batch, RJE
Third	Teletype timesharing
Fourth	Menu systems
Fifth	Graphical controls, windows

First generation: Knobs and dials

By this reckoning ENIAC and the tabulating equipment which preceded it were first generation systems—set up to solve specific problems by specialists with detailed and precise knowledge of the operation of the hardware. Many of the popular images of computers in the 1950s, seen in cartoons from the period, of the computer stretching from floor to ceiling and covered with knobs, dials, and oscilloscope screens, attended by mad scientists derive from the reality of first generation operation.

In the first generation, the user went one-on-one with the computer, in the computer room, operating the computer at the switch and knob level. Since the user was the operator of the machine and controlled it with little or no abstraction, there was essentially no mediation between the computer and its expert user.

Second generation: Batch

After ENIAC, virtually all general purpose digital computers followed the von Neumann architectural model and were therefore programmable without hardware reconfiguration. Even though until late in the 1950s most programming was done in machine language, requiring detailed knowledge of the hardware, the machine could be turned from task to task as rapidly as new programs could be loaded into memory. With computers built from vacuum tubes or discrete transistors being so expensive, extensive efforts were devoted to maximising the productivity of a computer, and as the 1950s waned the original model of computer usage, an individual user signing up for dedicated time on the machine, was supplanted by the batch shop, with a specialist computer operator running a stack of jobs. In time, computer operating systems automated much of the operator's work, assuming responsibility for scheduling, priorities, allocation of resources, and efficient management of the scarcest and most precious resources—CPU cycles and memory space. In time, the advent of remote job entry provided the same batch computing service available in the computer room to remote terminals located anywhere in the world.

The user's image of the computer during the second generation often revolved around a countertop. It was across the counter that the user handed the card deck containing his program and data, and across the same counter that, some time later, his cards would return, accompanied by a printout which he hoped would contain the desired result (but more often consisted of a cryptic error message or the Dreaded Core Dump).

Second generation operation introduced many important levels of mediation and abstractions between the user and the computer hardware. First and probably most important was the time shifting performed by a batch system and the autonomy this gave to the computer (or its operator) at the expense of the user's direct control. Since the computer did the user's bidding without an opportunity for user intervention, time limits, resource scheduling, recovery from unanticipated errors, and the like became a shared responsibility of the user and the autonomous computer operating system. This led to the development of *job control languages*, which provided a powerful (though often arcane) means of controlling the destiny of a task being performed by the computer without the user's involvement. The card deck, printout, countertop, and job control language form the heart of the user's view of a second generation system. The concurrent development of high-level programming languages reduced the degree of specialised knowledge needed to use such systems and made them accessible to more people.

Third generation: Timesharing

Throughout the second generation period operating system technology progressed toward the goal of squeezing more and more performance from computers. Early developments included *spooling* (from Simultaneous Peripheral Operation On-Line), which allowed a computer to process other work at full speed while devoting a small portion of its attention to running slow devices such as card readers and printers. Since many programs did not use the full capacity of the computer, but rather spent much of their time reading and writing much slower peripheral devices such as tape drives and magnetic drum and disc memories, operating systems were eventually generalised to allow concurrent execution of multiple jobs, initially in the hope of maximising the usage of scarce CPU and memory resources, and later with subsidiary goals such as providing more responsive service for small tasks while larger jobs were underway.

If a computer's time could be sliced or shared among a small number of batch jobs, why couldn't it be chopped into much smaller slices and spread among a much larger community of interactive users? This observation, and a long-standing belief that the productivity of computer users (as opposed to the productivity of the computer

itself) would be optimised by conversational interaction with the computer, led to the development of timesharing systems in the 1960s. Timesharing promised all things to all people. To the computer owner it promised efficient utilisation of computing power by making available a statistical universe of demands on the computing resource which would mop up every last CPU cycle and core-second. It promised batch users the same service they had before, plus the ability to interactively compose their jobs and monitor their progress on-line. And it offered interactive, conversational interaction with the computer to a new class of users.

Computer facilities were imagined, in the late 1960s, as agglomerating into regional or national "computer utilities", paralleling the electric power industry, which would sell computing capability to all who needed it, providing access wherever telephone lines reached, and all users a common database which could grow without bounds.

The archetypal device for computer interaction in the third generation was the Teletype model 33 KSR. It is hard to explain to people who didn't first enter computing in the batch era just how magical this humming, clunking, oil fume emitting, ten character per second device with the weird keyboard seemed. You could type in "**`PRINT 2+2`**", and almost instantly the computer would print "**`4`**". And most of all, you could imagine that device in your own home, linked by telephone to the computer whenever you needed it (the price of the hardware and the cost of computing in that age kept this a dream for virtually everybody, but it was a dream whose power undoubtedly contributed to its fulfillment, albeit through very different means).

The interactive character device, whether a slow printing terminal such as a teletype, or an ASCII "glass teletype" running at speeds of up to 960 characters per second, led to the development of *conversational computing*. The user types a line of input to the computer, which immediately processes it and responds with a reply (perhaps as simple as a prompt indicating it's ready for the next line). Many different flavours of this form of interaction were explored and coexist today, including the BASIC environment originally developed by Kemeny and his group at Dartmouth, editors such as TECO and its many derivatives such as VI, the project MAC timesharing environment whose influence is everywhere in the industry (including in Autodesk's own text editor), and eventually TOPS–10 and Multics (with their many derivatives including Unix and MS–DOS).

The conversational mode of interaction was the Turing test made real—the user "conversed" with the computer, just as he might with another human on a teletype-to-teletype connection (or CB on CompuServe today), and if the computer's responses tended toward comments such as "**`WAIT`**", "**`FAC ERR 400000044000`**", or "**`segmentation violation--core dumped`**" rather than observations about relevant passages in Wittgenstein, well with enough funding for the AI Lab and enough lines of Lisp, who knew?

Today it is fashionable to hold conversational computing in disdain, yet it achieved most of the goals that realistic observers expected of it. That it disappointed those whose expectations weren't grounded in the reality of what it was shouldn't be held against it—visionaries are always disappointed by reality (and therefore often lead the way to the next generation). In the guise of GE Timesharing, conversational computing introduced hundreds of thousands of students in high schools to computing, and many of these people now fill the ranks of the computing industry. The conversational model is almost universally the approach of choice by software developers, to the extent that Apple's own Macintosh Programmer's Workshop implements that model on a computer that is identified with another model entirely. The dominance of MS–DOS in the personal computer market and Unix in the technical workstation world (as well as the BASIC environment on most home computers, such as the Commodore and Atari 8 bit families) is testimony to the effectiveness and staying power of this mode of interaction. (It should be noted, however, that Unix in particular has in its shell programming facilities co-opted much of the second generation's job control languages, and a significant fraction of the power of Unix comes from integrating that approach with conversational interaction).

Fourth generation: Menus

Although conversational systems broadened the accessibility of computers, they still fell far short of the goal of making computers accessible to a large segment of the populace. As conversational interaction grew from slow (10 or 30 character per second) terminals, the appearance of fast alphanumeric terminals (1000 characters per second and up) made it possible to present large amounts of information to the user almost instantaneously. This allowed the computer to present the user with a "menu" of choices, from which selections could be made simply by pressing one or two keys.

Menu command selection, coupled with data entry modeled on filling in a form, rapidly became the standard for application systems intended to be operated by non-computer-specialists. Hundreds of thousands of people spend their entire working day operating systems of this design, although people who have studied how users actually learn and use these systems, in applications ranging from credit card transaction entry to targeting tactical nuclear weapons, often find that users see them in a very different way than the designers intended—frequently moving from menu to menu by rote learning of keystroke sequences, leaving the carefully-crafted menus unread.

Many attempts have been made to expand menu-driven systems into a general method of operation. Much of the Macintosh interaction model is actually fourth generation operation. Selecting commands from menus (whether presented directly to a user or pulled down from the top of the screen) and selecting options and setting program parameters by entering them in a form called a "dialogue box" is pure fourth generation design. The major point of departure from classic fourth generation structure in the Macintosh menu system is its attempt to place the user in direct command rather than treat the user as a peripheral who directs the computer as so many menu driven application systems do.

Fifth generation: Graphics

As monolithic integrated circuits began to relentlessly drive down the cost of computer memory, full screen raster graphics moved from a laboratory curiosity or specialised component of high-end systems to something which could be imagined as an integral part of every computer. Alan Kay and others at the Learning Research Group at the Xerox Palo Alto Research Center saw that this development, along with development of fast inexpensive processors, data networks, and object-oriented programming techniques, could lead to the development of totally new ways of interaction with computers. In the mid 1970s they explored the potential of these technologies on the Alto computer with the Smalltalk language. They involved children in their research program in part to take advantage of the unconditioned viewpoint a child brings to what he encounters.

Being able to express interaction with a computer on a two dimensional graphics screen allows many metaphors which can be only vaguely approximated with earlier technologies. The screen can be turned into a desktop, complete with pieces of paper which can be shuffled (windows), accessories (tools), and resources (applications). The provision of a pointing device such as a mouse allows direct designation of objects on a screen without the need to type in names or choose from menus as in earlier systems. This property has caused such systems to be referred to as *direct manipulation systems*. For example, file directories can be displayed as file folders on a screen, each folder containing a number of documents. If the user wishes to move a document from one directory to another, he need only grasp it with the pointing device and drag it from one folder to another. Compare this with the levels of abstraction inherent in the Unix command "**`mv doc/speech1.tex archives`**". This command, which quite explicitly specifies the same operation, requires the user to remember the name of the "move file" command is **`mv`**, the name of the sending directory, the name assigned to the file to be moved, and

the name of the receiving directory.

In addition, the availability of a graphics screen allows much more expressive means of controlling programs and better visual fidelity to the ultimate application of the computer. When editing documents, font changes can actually be shown on the screen. Controls which would otherwise have to be expressed as command names or numbers can be shown as slider bars, meter faces, bar or line charts, or any other form suited to the information being presented. Lee Felsenstein refers to the distinction between a conversational system and one like the Macintosh as the difference between a one- and a two-dimensional mode of interaction.

The extent to which five generations of user interaction with computers have brought us back to the starting point is ironic. Users of the first computers had dedicated access to the computer and direct control over its operation. The development of personal computers has placed the computer back in the user's hands as a dedicated machine, and event-driven interaction which places the user in immediate command of the computer's operation restores the direct control over the computer which disappeared when the user was banished from the computer room in the second generation. Use of graphics to express operating parameters is even restoring to computer applications the appearance of the computer control panels of the first generation, replete with meters, squiggly lines moving across charts, and illuminated buttons. This isn't to say we haven't come a long way—the meters on a Univac I console read out things like the temperature of the mercury delay line memories and the B+ voltage, and the switches allowed the user to preset bits in the accumulator. Today's displays and controls generally affect high-level parameters inside applications and allow the user, for example, to vary the degree of smoothing of a surface patch by moving a slider bar while watching a three dimensional shaded image change on the screen.

What next?

So, in the last forty years we've taken the computer user, who was initially in direct control of a dedicated computer, operating it by switches and gazing at huge arrays of blinking lights, to greater and greater distances from the computer and direct interaction with it, then back again to contemplating a virtual control panel on a glowing screen filled with slide pots, radio buttons, meters, all providing direct and expressive control over what's going on inside the computer. It appears that we've finally reached the end of the road—an individual has at his fingertips, for no more than the price of an automobile, dedicated computing power in excess of what existed in the world in 1960, with applications carefully tailored to provide intuitive control of the powerful tasks they perform, and a growing ability to move between applications at will, combining them as needed to address whatever work the user needs done.

It's interesting to observe the extent to which the term "user interface" has emerged as a marketing and, more recently, legal battleground following the introduction and growing acceptance of fifth generation user interaction. Many people would probably fail to identify anything before a fourth generation menu system as a "user interface" at all, though each generation has been how a user interacted, or in Eighties-speak "interfaced" with a computer system of that era.

Perhaps there's a semantic truth beneath the surface here. While one tends to speak of a "dialogue" or "conversation" when working with a line-oriented (third generation) timesharing system, only with fifth generation systems (and to a much lesser extent fourth generation menu systems) is one "face-to-face" with the computer. Maybe we keep referring to interfaces because we see our interaction as inter-face: our face dimly reflected in the screen that is the face of the computer.

I believe that conversation is the wrong model for dealing with a computer—a model which misleads inexperienced users and invites even experienced software designers to build hard-to-use systems. Because the computer has a degree of autonomy and can rapidly perform certain intellectual tasks we find difficult, since inception we've seen computers as possessing attributes of human intelligence ("electronic brains"), and this has led us to impute to them characteristics they don't have, then expend large amounts of effort trying to program them to behave as we imagine they should.

When you're interacting with a computer, you are not conversing with another person. You are exploring another world.

In *Computer Power and Human Reason* Joseph Weizenbaum spoke of this world, as seen by the programmer who creates it, as follows:

> The computer programmer is a creator of universes for which he alone is the lawgiver. Universes of virtually unlimited complexity can be created in the form of computer programs. Moreover, and this is a crucial point, systems so formulated and elaborated act out their programmed scripts. They compliantly obey their laws and vividly exhibit their obedient behaviour. No playwright, no stage director, no emperor, however powerful, has ever exercised such absolute authority to arrange a stage or field of battle and to command such unswervingly dutiful actors or troops.

This, Weizenbaum believes, explains the fascination programming holds for those who master it, and is the central reason why programming can become as compulsive a behaviour as any other activity that confers feelings of power, mastery, and pleasure.

The problem is that once a programmer has created a world intended for use by others, some poor user has to wander into it, armed only with the sword of his wits, the shield of The Manual, and whatever experience in other similar worlds he may have painfully gleaned, then try to figure out the rules. The timeless popularity of adventure games seems to indicate that at least some people enjoy such challenges, but it's much easier to exult in the discovery that the shiny stones cause the trolls to disappear when exploring the Cave of Befuddlement for fun than to finally learn that only if you do a preview will the page breaks be recalculated correctly for the printer when the boss is waiting for the new forecast spreadsheet.

If what's inside the computer is a world instead of another person, then we should be looking at lowering the barriers that separate the user from the world he's trying to explore rather than how to carry on a better conversation. Let's look at the barriers which characterise each generation of user interaction:

Generation	**Barrier**
First	Front panel
Second	Countertop
Third	Terminal
Fourth	Menu hierarchy
Fifth	Screen

Now there's little doubt that the greatest barrier between the user and the world inside the computer is the system designer's failure to comprehend that he's designing a world and to adequately communicate his vision of that world to the user. But also, as we've enriched the means of interaction between the user and the computer, both by raising the communication bandwidth and increasing the expressiveness of what is communicated by

using graphics, pointing, and the like, we've placed more powerful tools in the hands of the system designer to bring the worlds he creates to life and to involve the user in them. The fact that in the adventure game world the pure text line-by-line adventures remain the classics of the genre is an indication of how slow designers are to effectively exploit new capabilities.

Now we're at the threshold of the next revolution in user-computer interaction: a technology which will take the user *through the screen* into the world inside the computer—a world in which the user can interact with three-dimensional objects whose fidelity will grow as computing power increases and display technology progresses. The world inside the computer can be whatever the designer makes it; entirely new experiences and modes of interaction can be explored and as designers and users explore this strange new world, they will be jointly defining the next generation of user interaction with computers.

Through the screen to cyberspace

To move beyond the current generation of graphics screen and mouse, to transport the user through the screen into the computer, we need hardware and software that provide the user a three dimensional simulacrum of a world and allows interaction in ways that mimic interaction with real world objects. Several terms have been used to refer to a computer-simulated world, none particularly attractive. "Artificial reality" and "virtual reality" are oxymorons, Ted Nelson's term "virtuality" refers to a much more general class of computer worlds, "world simulator" is too grandiose for what we're talking about, and "cyberspace" misuses the root "cyber" (from $\kappa\upsilon\beta\epsilon\rho\nu\acute{\eta}\tau\eta\varsigma$—"steersman") to denote computer rather than control. Nonetheless, I will use "cyberspace" here to avoid burdening the discourse with still another term.[278] Since I'm talking about means of man/machine interaction, I can make the case that "cyberspace" means a three dimensional domain in which cybernetic feedback and control occur.

I define a cyberspace system as one which provides the user a three-dimensional interaction experience that provides the illusion he is inside a world rather than observing an image. At the minimum, a cyberspace system provides stereoscopic imagery of three dimensional objects, sensing the user's head position and rapidly updating the perceived scene. In addition, a cyberspace system provides a means of interacting with simulated objects. The richness and fidelity of a cyberspace system can be extended by providing better three dimensional imagery, sensing the user's pupil direction, providing motion cues and force feedback, generating sound from simulated sources, and further approximating reality almost without bounds (wind in the face, odor, temperature—direct neural interface, anyone?).

The idea of transporting the user in some fashion into a computer and allowing him to interact directly with a virtual world has been extensively explored in science fiction. Frederick Pohl's later Heechee books (the second of which, *Beyond the Blue Event Horizon*, inspired the product Autodesk, which idea played a significant part in the formation of this company), writers of the "cyberpunk" genre such as William Gibson and Rudy Rucker, and movies including even *Tron* have explored what we will find and what we will become when we enter these worlds of our own creation. It's no wonder the idea of *entering* a computer world is so fascinating—it can be thought of as the ultimate realisation of what fiction has been striving for since sagas of the hunt were told around Paleolithic campfires. The images that prose and poetry create in the mind, that the theatre enacts on stage, that motion pictures and television (aided by special effects) bring to millions can, inside a computer, not

[278] In retrospect, this was a bad choice for a name, since in the years which followed "cyberspace" has come to mean the global web being woven through the integration of telecommunications and computing, and the Internet in particular. But in 1988, I had no way to know this.

only be given three-dimensional substance, but can interact directly with the viewer, now a participant rather than passive spectator.

Science fiction's attention to the idea of entering a computer world should not be taken as an indication that the idea is itself infeasible, part of the distant future, or a fictional device devoid of practical applications. Ivan Sutherland, who invented so much of what we now consider commonplace in the computer graphics industry, realised in the 1960's that using two small CRTs to provide stereoscopic images to the eyes and sensing head position to compute the viewpoint was the way to three dimensional realism. In 1968 Sutherland built a helmet with two CRTs, attached to the ceiling with a set of linkages and shaft encoders to determine head position.[279] This contraption, called the "Sword of Damocles" because of all the hardware dangling above the user's cranial vault, really had only one serious flaw—it was twenty years ahead of its time in the computer power required to make it practical.

Now that fast CPUs and special-purpose graphics hardware have made real-time generation of realistic 3D images widely available at reasonable cost, and every expectation is that the ongoing trend of increasing performance at decreasing cost will soon bring that power to personal computers, the technological groundwork is in place to bring Sutherland's prototype into the mainstream of computer graphics. Interest in head mounted displays and other cyberspace technologies is growing. Attached to this paper are an article from the October 1987 *Scientific American* titled "Interfaces for Advanced Computing" which surveys the field and describes current technology, a paper titled "Virtual Environment Display System" by the group at NASA Ames who built the first modern cyberspace system, an article from the August 15th, 1988 *Aviation Week and Space Technology* describing a helmet-mounted display with head tracking being tested by Navy aviators, and an article from the August 22nd issue of the *Independent Journal* which indicates that the potential of cyberspace is beginning to filter down to even minor suburban dailies.

Building cyberspace

Exploring cyberspace requires specialised hardware and software. The attached articles describe the hardware used in the current laboratory systems; please refer to them for details. Here's an overview of what constitutes a minimum contemporary cyberspace system which can be assembled from off-the-shelf components and a sketch of the outlines of a complementary software project to initially explore the applications of cyberspace to our products.

Cyberspace Hardware

To provide the illusion of being within cyberspace, the system should provide a stereoscopic image that tracks head position. A system that does this can be built by using two small video monitors mounted on a helmet the user wears. Affixed to the helmet is a head-tracking device, such as the Polhemus Navigator (made by a subsidiary of McDonnell Douglas) which provides eighth inch position and quarter-degree angular accuracy without attached wires.

The video displays can be fabricated from components salvaged from LCD pocket televisions, or camcorder viewfinders can be used as-is. (The current NASA design uses custom displays and optics to achieve a wide

[279] Sutherland, I. E., *A Head-mounted Three Dimensional Display*, Proceedings of the Fall Joint Computer Conference 1968, Thompson Books, Washington, D.C., pp. 757–764.

field of view, but their first prototype used commercial LCD displays). Each monitor is attached to a separate graphics controller which renders the view of the three dimensional model of the world from that eye's viewpoint, updating the display as the head translates and rotates. Cyberspace experimenters stress that quickly updating the display as the viewpoint changes is essential both to maintain the illusion of being in a simulated world and to avoid vertigo. NASA has found that a fast wireframe display is preferable to shaded imagery that lags head movement.

One could configure an initial experimental cyberspace system using, for example, two Amiga 500 home computers as rendering engines, fed a three dimensional model created with AutoCAD or AutoSolid by a control computer (either a Compaq 386 or a Sun would work well) which monitors head position and sends viewpoint updates to the rendering engines.

For user interaction with the cyberspace environment, one could use the VPL glove which, with available software, allows recognition of commands from hand gestures and, with a Polhemus navigator attached to the glove, pointing and grasping of objects in cyberspace. Other input devices such as joysticks and foot pedals could also be explored. The entire hardware complement needed for this initial cyberspace exploration and demonstration system would cost less than $15,000 (not counting the control computer, which would not be dedicated to the system in any case) and could easily be transported and set up wherever required.

This system is so simple and transportable that I call it "cyberspace in a briefcase". It would serve as an initial prototype to demonstrate the value of cyberspace environments, and introduce our user, developer, hardware manufacturer, and analyst communities to the potential of our work in the area. Our goal in assembling an initial cyberspace system and demonstrating its potential would be to spur the graphics hardware manufacturers who work closely with Autodesk into cooperating with us to specify, develop, and market commercial cyberspace hardware to work in conjunction with cyberspace software developed by Autodesk. Manufacturers of high-performance graphics peripherals are often disappointed that Autodesk does not push their products to the limit. Improving the realism of cyberspace systems can use all of the capacity of the next several generations of graphics hardware (while being useful even with currently affordable products).

Cyberspace Software

For initial explorations of cyberspace, software should consist of a toolkit which allows rapid prototyping of cyberspace environments. Because cyberspace is so new and the fundamentals of how one should interact with it remain to be discovered, we should attempt to prescribe as little of the interaction as possible in the toolkit itself, but make it easy for those who use the kit to define their own environments.

Autodesk has a large advantage in undertaking cyberspace software development. A cyberspace environment is a three-dimensional computer model, and Autodesk has a rich set of off the shelf tools for building and manipulating 3D geometry. As a result, the amount of software development specific to cyberspace will be limited—AutoCAD and AutoSolid can be used for modeling without modification, and appropriate AutoLisp routines and "glue" can be developed to create an effective utility belt for the cyberspace explorer.

Cyberspace Environments

So, what does the world look like to the intrepid cybernaut? Whatever he wants! Cyberspace is unlimitedly rich because it can be anything at all. In time we may expect that conventions for cyberspace will evolve, just

as they have for command line, menu, and graphical user interfaces, but cyberspace will always provide an arena where anything that can be imagined can be made to seem real.

Initial cyberspace environments will literally represent three dimensional models. Since cyberspace is the most natural way to work in three dimensions, we expect that three dimensional design will be the first major application area for cyberspace systems. But as William Gibson says, "The street finds its own uses for things". Just as AutoCAD has been applied to many tasks well outside the traditional bounds of the "CAD market", cyberspace can be expected to rapidly grow in unanticipated directions. If video games are movies that involve the player, cyberspace is an amusement park where anything that can be imagined and programmed can be experienced. The richness of the experiences that will be available in cyberspace can barely be imagined today.

Menus might be replaced by doors you walk through to enter new worlds (certain doors would be unlocked by the key of imagination). A **ZOOM** command could be implemented by grabbing the appropriate mushroom—one makes you larger, the other makes you small. Need **HELP**? Go ask Alice.

As conventions develop for defining cyberspace environments, cyberspace will be applied in increasingly abstract ways. A cyberspace system may turn out to be the best way to implement a hypertext browsing system, or for visualising scientific data in multidimensional space (one could imagine a "transdimensional cyberspace Harley" that lets you ride along any vector in the state space).

In designing interactive systems we must distinguish abstractions introduced because of the limitations of the medium (for example, abbreviations to compensate for a slow teletype) and abstractions that add power or intuitiveness to the interface (such as the ability to create macros to perform repetitive tasks). By creating a very rich environment, cyberspace allows us to dispense with the abstractions of compromise and explore the abstractions that empower the user in new ways.

Should Autodesk lead?

If cyberspace is such an obvious next step in user/computer interaction, then it's reasonable to ask why Autodesk should expend any effort to develop the technology in-house. Can't we just let others do the pioneering and adopt their discoveries as they reach the market?

I think that Autodesk should be a leader in making cyberspace a mainstream technology because Autodesk has several attributes which uniquely qualify us to develop cyberspace. I believe that Autodesk stands to benefit enormously if we are successful in developing the technology and bringing it to market in conjunction with our product line.

Autodesk's business

Cyberspace is a general purpose technology of interaction with computers—nothing about it is specific to 3D graphical design any more than fifth generation interfaces based on raster graphics screens are useful only for two dimensional drawing. New technologies, however, tend to be initially applied in the most obvious and literal ways. When graphics displays were first developed, they were used for obvious graphics applications such as drawing and image processing. Only later, as graphics display technology became less expensive and graphics displays were widely available, did people come to see that appropriate use of two dimensional graphics could help clarify even exclusively text or number oriented tasks.

So it will be with cyberspace. Cyberspace represents the first three dimensional computer interface worthy of the name. Users struggling to comprehend three dimensional designs from multiple views, shaded pictures, or animation will have no difficulty comprehending or hesitation to adopt a technology that lets them pick up a part and rotate it to understand its shape, fly through a complex design like Superman, or form parts by using tools and see the results immediately. Those who had to see shaded pictures to appreciate the value of rendering software and experience their first fly-through to consider animation something more than a gimmick are sure to appreciate cyberspace only after they have stepped into it the first time.[280]

Since Autodesk's business is three dimensional design, we not only have the tools needed to build cyberspace environments as the heart of our product line, we have as customers the most likely early adopters of cyberspace systems—the pioneers in applying cyberspace to their application areas.

Autodesk's technological leadership

I believe that cyberspace is the only technology which is a serious contender to define the next generation of user interaction. Today, Autodesk is faced with the challenging task of finding effective ways for users to manipulate three dimensional models while conforming to the conflicting standards of numerous competing fifth generation graphical interfaces. This job is made difficult because of the lack of standards and by the inherent difficulty of trying to work on a 3D problem through a 2D window.

If Autodesk establishes itself as the leader in exploring cyberspace, and is forthright in identifying its effort as explicitly attempting to invent the next generation of user interaction, we will to a large extent transcend the quibbles over conformance with the last generation of user interface standards (which is not to say that we can ignore them, nor that Autodesk need not continue to upgrade our products' interfaces in conventional ways). The cyberspace project will be a technology flagship, demonstrable in all forums in the near term, which will clearly position Autodesk as a leader in technology and innovation in our core business, 3D design, just at the time when our marketing effort will be aimed at making Autodesk a peer of the big CAD companies. For years those companies have been defending their territory by enumerating things that we couldn't do. Isn't it time we simply superseded them by developing the next generation of interaction and causing it to be identified with our products just as strongly as the fifth generation interface has become identified with Apple?

Autodesk's future

As cyberspace systems mature they will redefine the way products are designed and operated just as thoroughly as have graphics screens and mice. If we take advantage of our opportunity to lead in the development of cyberspace, then our products will be the first to incorporate it, our users the first to explore it, and our applications developers the first to apply it in the many industries they serve. This will give us a huge head start in building the first products of the age of cyberspace—the products which will define the common ground of interaction in that space and from which all latecomers will be forced to somehow distinguish themselves.

Our experience in the PC CAD market should have taught us the value of arriving first in an empty market. Our experience in playing catch-up in the 3D world and in upgrading AutoCAD's user interface is a testament to how difficult it is when you don't get there first. If cyberspace truly represents the next generation of human

[280]The NASA Ames group says that one of the problems in demonstrating their system to Congressbeings is that they can't get the Distinguished Lawmakers to take off the cyberspace helmet—they keep asking to fly through "just one more model" and would spend all day in the cyberspace lab if not dragged away by the tour guide.

interaction with computers, it will represent the most profound change in the industry since the development of the personal computer. By helping to bring about that change, Autodesk can emerge as one of the few key players in the next phase of the industry's expansion.

Autodesk's investment

The investment required by Autodesk to explore cyberspace interaction is modest, comparable in size and expense to most other in-house product development efforts. A group of three or four programmers should be able to demonstrate a cyberspace environment within two to three months after project inception, with capabilities added to the system, interfaces built to existing and new products, and adaptation of the system to new hardware systems progressing as work continues after the initial system is demonstrated.

The initial experimental system would be built by cobbling together off the shelf hardware, probably engaging the services of a hardware consultant to help us assemble the gizmo. After this initial system was built (I believe that $25,000 is plenty of money to fund its construction), development would focus on software designed for easy portability to new hardware as it became available. Autodesk would use the initial experimental system to interest hardware vendors in working with us on cyberspace technology. After we demonstrate what we can do with the crude original system, I suspect that we will have no problem finding vendors eager to develop more powerful, professional, and inexpensive solutions to the problems of cyberspace interaction. Once again, Autodesk's preexisting close relationships with hardware manufacturers give us a large advantage over others in promoting this technology.

A project of this scale should begin to yield deliverable results, both new products and cyberspace additions to our existing products, in about a calendar year after inception.[281] Because of the necessity of involving hardware companies in the project and the time it will take to explore the potential of cyberspace before beginning to design products, I don't believe a larger project would yield results any faster.

Autodesk's opportunity

The history of the computer industry consists of the realisation of dream after dream initially dismissed as "only science fiction". The ability to place users in computer-generated three dimensional environments and allow them to interact with simulated objects will begin to break down the barrier between the user and the world inside the computer. This may usher in totally new ways to interact with computers, new applications for computers, and even new ways of thinking about computers.

Twenty years after Sutherland's prototype demonstrated the feasibility of computer simulated realities (cyberspace), the technology required to build cyberspace systems is almost in hand and can be expected to become widely available at low cost within the next several years. Development of cyberspace systems, driven by their near-term practical applications in control of complex systems, teleoperation, rapid presentation of information in aerospace environments, and more effective manipulation of three dimensional models, is underway in several laboratories and will soon move into the marketplace.

Autodesk is well positioned to be a leader in this new industry. Our fundamental core business is three

[281] The Cyberspace project was launched not long after this paper was circulated, and Eric Gullichsen produced a working prototype within 90 days. The actual product, the Cyberspace Developer Kit, a C++ class library for building and manipulating virtual worlds, shipped in February 1993, about four years after the project was launched. Oh well.

dimensional modeling. Our products stand to benefit from having the most effective three dimensional interaction of any vendor's, and from being correctly seen as leaders in making the complex tasks they do widely accessible. Our relationships with application developers and hardware manufacturers permit us to cooperatively develop this technology without having to bear all the costs and risk.

Autodesk's decision

Autodesk can identify itself with one of the most exciting developments ever to happen in the computer industry, one which may define the entire shape of the industry for the next decade. I believe that effort required to do this is on the order of that we expend to bring a new product such as AutoShade or AEC Mechanical to market. We should have no difficulty in finding people interested in developing this technology and no problem promoting it once we can demonstrate its potential.

If Autodesk believes, as I do, that this technology not only holds the key to the next generation of user interaction but will first find applications in our central market, three-dimensional design, then Autodesk should apply resources to developing this technology commensurate with its potential. If we undertake this project, we should commit to it explicitly and allocate adequate manpower to get it done. If the project merits only the efforts of one burned-out programmer in his spare time, then it isn't worth undertaking at all. An "Autodesk Cyberpunk Initiative" which will yield results within four months and products within twelve is affordable, achievable, and appropriate.

Autodesk can pioneer cyberspace. We need only the vision to see the opportunity, the courage to break new ground, the decision to do it, and the will to see it through.

Cyberspace

Reality isn't enough any more.

AutoCAD Expo Moscow

Autodesk was actively doing business in Eastern Europe and the Soviet Union long before it was fashionable to do so. The opening of AutoCAD Expo Moscow on October 17, 1988 marked a major milestone in developing the market for AutoCAD there. It was the first software-oriented trade show ever organised in the Soviet Union by a western firm. Richard Handyside, who led the effort to open the then "Eastern Bloc" for Autodesk wrote this summary as soon as he arrived back in London.

Date: Mon Oct 24 13:40:16 1988
From: Richard Handyside — Managing Director Autodesk UK
To: Autodesk employees worldwide
Subject: AUTOCAD EXPO MOSCOW 1988

This is just a quick preliminary report on AutoCAD Expo Moscow, which we got back from on Saturday night.

The Expo was a tremendous success from every point of view—for Autodesk, for the 19 developers and dealers exhibiting with us, and for the 8000+ visitors who attended what was the first-ever international software exhibition in the Soviet Union.

Response to the Russian-language version of AutoCAD Release 10, which we launched at the show, was overwhelming. It was physically impossible to get through the crowd around the machine it was being demonstrated on for most of the show.

Interest in AutoSolid was equally overwhelming—literally, physically so most of the time: one needs to evolve a whole devious technique for bringing out a new box of literature if one doesn't want to get crushed in the mob at one of these shows. Our Mr Solid, Bill Barnes, talked himself hoarse, and had to resort to French when English didn't work. At the end of the show, we announced that we would be donating the first copy of AutoSolid in the USSR and the first copy of the Russian AutoCAD Release 10 to AZLK, the Moskvich car factory who are licenced AutoCAD and AutoShade users and who gave us tremendous help in organising the Expo in their conference centre.

Response to the seminar program was enthusiastic—so much so that the crowd trying to get in to hear the opening Autodesk seminar on the first morning smashed the auditorium door off its hinges! And it was exciting to walk through the exhibition and overhear Russian visitors saying to each other what a great show it was. Among the more interesting visitors was Vitaly Sevastanyov, a Soviet astronaut who has been up in space 3 times and is an academician of the World Academy of Astronautics.[282] His presence wasn't mere PR, either:

[282] Sevastanyov was manager of the Soviet lunar mission training group in 1967, and flew on Soyuz 9 in 1970, Soyuz 18/Salyut 4 in 1975, and a later mission I've failed to find in my Soviet space program references. —JW

he came to the Expo and attended seminars 3 days in a row.

The exhibition was a success in terms of immediate business. We, Autodesk, signed contracts there for over £75,000 (US$128,000): we sold 30 copies of AutoCAD, 2 copies of AutoSolid, and 3 copies of AutoShade. Including one exceptional hardware+software contract for £1.2 million, the total value of contracts signed during the show came to nearly £2.2 million (US$3.75M). And at least as many contracts again did not get concluded during the show but will follow shortly.

These are just the immediate results. The Expo will without question mark a decisive turning-point in our activities in Eastern Europe in general. We signed up 3 very important new dealers, among them a Soviet-American joint venture company called Dialog, who are also going to be licenced distributors for Microsoft, Ashton-Tate and Lotus. The Union of Architects, covering all architects in the USSR, has decided to standardise on AutoCAD: they will shortly be setting up an AutoCAD Authorised Training Centre, and also plan to sell complete AutoCAD systems. The East German Bauakademie (Construction Academy) will be signed up, once they have completed their internal approvals, as an ATC, support centre and applications developer: this academy has 4000 engineers on staff and is responsible for all construction in East Germany.

A tremendous high for everyone involved: now we have to settle down to the huge amount of follow-up work that is needed. Visitors, exhibitors and the AZLK car factory all asked when we would hold the next Moscow Expo. The answer is October 1989, by which time we confidently anticipate that there will be thousands of legal, licenced copies of AutoCAD and AutoSolid in productive use in the USSR—and a whole flowering of exciting new applications programs.

Susan Sheridan[283] came over with us, and with her vigorous arm-waving Polish was a tremendous help to the 5 of us from the Autodesk London office: she'll no doubt have her own stories to tell. One of the exhibitors recorded a lot of the Expo on video, and we'll send over a copy of that for Susan to show asap.

Another great first for Autodesk.

Richard

[283] Marketing Director at Autodesk Sausalito.

Golden Hammer

The shipment of AutoCAD Release 10 in October 1988 marked a major milestone for Autodesk. After years of smarting from the claim "Autodesk can't do 3D", we finally shipped an AutoCAD which not only contained genuine 3D, but surfaces, multiple on-screen views, perspective viewing, and dynamic viewpoint selection.

This was the culmination of the Scott Heath's 3D project, launched twenty months before at the programmers' meeting at my house (see page 382). I thought some special recognition for Scott was in order at the next monthly company meeting/beer-bust. Dusting off a tradition I'd let lapse since the 1970's, I went out and bought a wood plaque, a sledgehammer, and a can of gold paint, and set to work. The following was a total surprise to Scott. This is, to date, the only Golden Hammer awarded at Autodesk, although Heaven knows there are dozens of people deserving of one.

Statement for the Autodesk Monthly Meeting

by John Walker — November 4th, 1988

By the end of 1986, most of the developers of AutoCAD, including many of the founders of the company, despaired of extending AutoCAD into a full three-dimensional CAD system.

"We'll need to redesign it."
"A total rewrite is the price of admission."
"We're getting nowhere—maybe this is the miracle we won't be able to pull off."
"We may be able to make a 3D CAD system, but it won't be AutoCAD any more."
"It'll *never* fit in 640K."

And all the time we reached for the strength within ourselves to build a 3D AutoCAD, our competitors made the most of the claim that "Autodesk can't do 3D".

One voice argued otherwise.

One person said, calmly but insistently, "Adding 3D to AutoCAD is a programming project just like all the others we've done. It's a big job, so let's get started on it right now."

In March of 1987, the development of AutoCAD "Abbey Road"[284] got underway in earnest.[285] The outcome of that effort, without question the largest and most difficult job of software development ever undertaken at Autodesk, is now shipping as Release 10, and Autodesk is in the 3D business to stay.

Now I don't like to put people on the spot, and in a project as sprawling and protracted as the development of Release 10, not to mention everything else we had to do before we shipped it, it's hard to thank everybody without rolling something like movie credits.

But... if we fail to honour the heroes among us, we deny that which is the noblest part of ourselves.

Release 10 is the product it is today, and is shipping to our customers, and is upwardly compatible with Release 9, and has multiple windows on screen, because of the vision of Scott Heath.

That vision and the willingness to see it through month after month of coding, debugging, overlaying, squeezing, debugging again, performance tuning, debugging yet again: carrying that weight a long time, has brought the vision to reality. That reality is Release 10, and it is the best AutoCAD this company has ever shipped.

In one of my former lives, I used to publish an underground newspaper called the *Shifting Bit*. One of the highlights of the paper was a contest I ran every year to identify the programming project that best exemplified the rule, "If it doesn't fit, get a bigger hammer".

The winner received the Golden Hammer. I last awarded the Golden Hammer in 1976.

If ever any project deserved the Golden Hammer, AutoCAD Release 10 does. Not only does it manage to squeeze a full 3D CAD system with surfaces into the same 640K that held Release 9, it even has perspective and multiple windows on screen.

Scott, you win. I'm honoured to present you with the *Shifting Bit* Golden Hammer award for 1988 for seeing how to get from AutoCAD Release 9 to Release 10, for leading us there, and for hammering on that sucker 'till it fit in a PC.

AutoCAD Release 10 marked more than AutoCAD's maturation from an extruded $2\frac{1}{2}$D adolescent into a sinewy compound curved 3D adult; it also heralded the passing of the cattle prod of AutoCAD development from the founders, the central figures in the development of AutoCAD from its inception through Release 9, to a new generation of developers. Not only do these folks possess knowledge of CAD, graphics, and geometry that dwarfed that of the founders of Autodesk, they have the energy to tackle daunting tasks of programming and the courage to try again where we had failed and given up. Release 10 stands as evidence that AutoCAD is in the best of hands.

Indeed, AutoCAD Release 10 was the first mission of "Autodesk, the Next Generation": and what a success it has been. Now, as we look forward to AutoCAD Release 11, and then Releases Twelve through Twenty, we can look to the people who gave us Release 10 with calm confidence in the future and eager anticipation of the wonders to come.

Co-ordinates: The Golden Age of Engineering—Warp Factor: 11—Engage....

[284] After Release 9, "White Album," the choice of a code name for Release 10 was obvious. After Release 11, "Let It Be," (particularly apt since Release 11 was to be the first release which would not run on 640K DOS [we did, subsequently, make a 640K DOS version of Release 11]) we ran out of Beatles' albums, creating a crisis in code names. (No, I don't consider "Let It Be" to be a genuine Beatles' album either, but let's take that debate off-line.)

[285] Kicked off at the programmers' meeting at my house. See page 382.

One Hundred Million Dollars

Autodesk achieved its most recent power-of-ten milestone in late 1988, when sales for that year passed the one hundred million dollar mark. Things sure had changed since the party we held in January 1984 to celebrate our first million dollar year! (See Information Letter 11 on page 227.)

For a milestone ten times as large, the celebration was far more muted; we held a brief meeting at 5 P.M. in the beer bust room in Sausalito. We had nine binary cakes, a one and eight zeroes, to celebrate the sum. (Actually, there may have been a tenth cake with a dollar sign; I'm not sure.)

A mere three years later, in the fiscal year that ended in January of 1992, Autodesk's sales in Europe alone *exceeded* $100 *million.*

Statement for the $100 Million Meeting

by John Walker

December 13th, 1988

One hundred million dollars. Now there's a number to ponder.

I looked back in the AutoCAD development log to see where we were on this date in 1982. As far as I can tell, we managed to exceed $100 million in sales six years to the week after we shipped the very first production AutoCAD.

And you people have nobody to blame for this but yourselves. Because there's nothing here, really, but people. It used to be you had to have a big factory, a fleet of buses, or stores all over to build a $100 million business. Autodesk has done it with imagination, information, hard work, and the best people in the business. All of the source code, all of the masters, all of the documents for all of our products will fit on this little tape. Everything else that this company is and will become is in the people who did it—the people here, the people who couldn't make it, the people overseas, and the people who contributed to our success and have since moved on.

What does a hundred million dollars in sales represent? If you look at the total you miss the reality. The reality is that all the money came from building the best products, then manufacturing, selling, shipping, testing, supporting, marketing, training, accounting, administering, and doing all the innumerable and often invisible and thankless tasks that turn a good idea into a great business.

The problem with these "power of ten" parties is that it's so hard to get to the next one. When we did $1.4 million in sales in our first full year we had a celebration that was so memorable I don't remember very much

about it, but I do recall saying that we'd all get back together some day to celebrate $10 million. I don't know how many people took that seriously. I did.

Well, here we are celebrating $100 million. Am I going to stand here and seriously talk about a billion dollar party?

Yep.

How do we get there? Exactly the way we got here, by working away at all the tasks that need to be done, worrying about the details, trying to do the right thing as we see it, and never losing the imagination that is the only chart to an unknowable future in a dangerous world.

What will Autodesk become in the years to come? Precisely what we make of it. And the work we do tomorrow and tomorrow and tomorrow will carry us from this happy milestone to the next.

The New Technological Corporation

After relinquishing the chairmanship in mid-1988 (see pages 420 and 422), I had both a lot more time to think and to worry about how the company would be run in the future. From conversations with folks in the senior management and the occasional management meeting I attended, it became very clear to me that frequently decisions were made which seemed intuitively wrong, but when I tried to explain why I would have acted differently, I could not convey my argument in terms the management could understand.

This caused me to probe the theoretical foundations that underlay my intuition. I figured that if I could explain them*, then I'd equip my audience with the intellectual tools I'd been using, subconsciously, over my career; whether they chose to use them, of course, was their business and not mine. As I wrote this paper, I was a little surprised at just how closely such apparently unrelated threads are woven.*

The paper had, as far as I can determine, no impact whatsoever upon anything. As far as I can tell nobody ever read it and, if they did, failed to understand it or though it was only about dividends, rather than how software companies should be managed to exploit their unique advantages.

This is the theory of management which guided the foundation and growth of Autodesk, and it forms the basis of the analysis and recommendations I wrote more than two years later in Information Letter 14 (see page 600).

The New Technological Corporation

The software business differs in fundamental ways from established industries. Viewing the economic fundamentals of our business in the context of the larger economy suggests Autodesk may be an exemplar of a new class of information-intensive companies.

by John Walker
December 21st, 1988

One phrase that seems to recur when explaining the software business to people familiar with other industries is "this isn't like any other business". This assertion is often dismissed out of hand and, indeed, many aspects of the software business are analogous to other long established and readily-understood businesses.

But the software business is unique. The combination of minimal capital spending requirements, the extremely high operating margins that result from successful software products, low up-front investment to develop and launch a product, and the inability to predict which products will succeed among a large number of potential products combine to define an industry which behaves, both as seen by management inside and by analysts and investors examining operating result aggregates outside, unlike any well-known model.

If the software business is fundamentally different from, say, book publishing, semiconductor manufacturing, financial services, or management consulting, one cannot look to those sectors to provide prototypes of how a software company should be organised, managed, grown, and valued in the capital markets.

This paper will start from first economic principles to examine the fundamentals of a software company. Viewing those fundamentals as part of the overall economic system suggests that software companies are unique; indeed, the software industry may be the exemplar of a new class of information-intensive businesses, the New Technological Corporations. These companies must find their own way to the strategies that best fit the realities of their business. Operating experience, growth strategies, and principles of valuation derived from high-technology hardware manufacturing may be no more relevant to these new industries than the experience of building a railroad would be to broadcasting.

Theme 1: The time value of money

> *No number is as central to economic decision making as the discount rate, the numerical expression of the value of a coin in the hand versus a billfold in the bush. The risks and rewards of all investments have meaning only when compared to the prevailing discount rate.*[286]

First prelude: The age of takeovers

Recent years have seen the emergence of corporate takeovers on a scale unprecedented in the history of commerce, whether measured by the number of transactions, the total size of the deals, the audacity of the raiders, or the creativity of the instruments devised to finance the acquisitions.

Why is this? Why should corporations embark on a binge of devouring one another when equity market valuations are close to all-time highs measured by earnings and dividend yield, when interest rates on the debt used to finance takeovers are at historically high values, and why should these takeovers continue unabated after the worst stock market crash in history and the advent of a primary bear market marking the end of the longest period of economic expansion in decades? Are we witnessing an epiphany of Mammon where greed and rapacity trample reason, or is there an underlying rationale for these deals here, now? I believe there is, and that it illustrates the central importance of the time value of money.

When a "corporate raider" solicits the owners of a corporation to tender their shares at above the prevailing market price he is, in essence, saying that he disagrees with a valuation for a body of visible assets arrived at by

[286]The discount rate is an economic measure of the pure time value of money. Frequently news media use the term "discount rate" to denote the Federal Reserve Rediscount Rate, the rate the Federal Reserve charges banks to borrow short-term funds. This number has little to do with the true prevailing discount rate, which is set by the expectations of borrowers and lenders about the supply of and demand for money, the prospects for business, expectations of inflation, and a host of other factors.

the largest, most efficient, market ever created. Why do so many people who are wealthy beyond imagination have the audacity to dismiss the judgement of the marketplace? Ego?... greed?... or something else?

Perhaps the takeover artists are not the cause, but the effect, of a historic imbalance in the time value of money.

Reinvestment and risk

At the heart of the concept of the corporation is the assumption that it will generate profits (or savings), and dispose of them in the best interests of its owners, the shareholders. In the absence of taxation, management would determine what percentage of earnings should be reinvested in the corporation to maintain its position in the market and take advantage of opportunities for growth and competitive advantage, versus what should be paid out to the owners as compensation for the capital they have contributed to the corporation by purchasing its stock.

Structures of taxation which treat corporate earnings, individual income, debt service payments, and dividends differently shift the optimum strategy. Uncertainty regarding future tax policy and time lags while market participants adjust their strategies in the face of changes in taxation further complicate the process of arriving at optimal strategies. Nevertheless, taxation at the levels currently obtaining in the West affects the key decisions in deployment of corporate resources only on the margin (except for the double taxation of corporate earnings paid out as dividends; this will be discussed in greater detail below).

If the management of a corporation were omniscient, discharging their fiduciary duty to the shareholders would entail calculating the future gains to be realised by retaining corporate earnings and spending them to further develop the corporation, versus rebating them to the shareholders so that the funds may be invested as the shareholders see fit, presumably with a return no less than the discount rate—the zero risk time value of money, for which the short-term government security in a given currency serves as a proxy.

An underlying assumption of stock market investment has always been that the management of a successful corporation in a viable market could be expected to reinvest earnings in their business with an eventual yield greater than that of risk-free investments. In other words, management's knowledge and the position of the company in the marketplace will result in an expected yield that exceeds the return of zero-risk short-term debt instruments. Were this not the case, what would induce investors to forsake risk-free investments to entrust their funds to a venture where, in the direst extreme, they could lose all of their capital? For taking a chance on the future of the company, the investor demands a "risk premium"—greater total return, on the average, as compensation for assuming the risk.

In recent years an historic reversal of this situation has occurred, and it is at the heart of the takeover boom and the slow-motion liquidation of many long-established corporations. The historical discount rate for long-term (e.g. 30-year) money over the last several hundred years has been between 2% and 3%. Consider an oil company which long-term experience indicates can invest in wildcat exploration combined with an ongoing drilling program in established oil-bearing leased areas to yield a 6% contribution to future earnings from funds committed to exploration and development. Were the long term government bond yielding 3%, most investors would gladly endorse this investment in the future value of their shares, as the potential gain would be twice that of the risk-free alternative.

As I write this paper, the interest rate paid by risk-free 90 day United States Treasury Bills is 8%—more than two and a half times the historical discount rate and a third again more than the yield that the best managements have obtained by reinvesting in their businesses in the last two hundred years. Why, then, do managements

continue to drill oil wells, fund superconductivity research, launch new brands of deodorant, devise new ways of delivering sugar to the children of America, and otherwise contribute to the common wealth of humanity? I suggest it's because they don't know what else to do.

If you're an oil man, you drill, even if others snicker as your instinct becomes obsolete in an age where reserves can be purchased cheaper on the open market than sought by exploration. If you're a soap man, you try to find a new niche for the soap to wash expensive athletic sneakers.[287] If you're a cereal murderer, you seek new ways to package and promote white sugar, and so on.... And do you ever think about the alternative of just buying a Treasury Bill or giving the cash back to the shareholders? Well... no.

But somebody does—the "corporate raider"—that's what he's paid to do. His economic function is performing arbitrage between the returns to be had by reinvestment in a company's business versus liquidation and return of capital to the shareholders.

The invisible hand and its elbow

Now that we've thought a bit about reinvestment in a corporation as opposed to return of earnings and capital to the shareholders, let's look at a Leveraged Buy-Out (LBO) transaction as the consequence of this calculation by the shareholders. Many companies taken private by LBOs seem, groaning under the burden of servicing the debt undertaken in the buy-out, perched on the "lap of God". I refer to the process that puts them there as the "elbow of the invisible hand".

Imagine a successful company in a stable, easily-analysed industry with pre-tax earnings of 12.3%. If that company has no interest payments or other significant deductions, it will pay about 35% in corporate income tax on its earnings, leaving about 8% net savings. If the corporation pays out 3.8%, the average dividend yield on the Dow Jones Industrials at this writing, about 4.2% is left for reinvestment in the business. If the economy and Fortune shine on the judgement of management, and this reinvestment doubles in a year, the gain in earnings will be 8.4%, a princely sum by the standards of history, but little more than the riskless interest to be earned by the simple expedient of purchasing a Treasury Bill. So consider the plight of an investor in this company. He places his capital at total risk, subject to loss not only from incompetent management and competition, but also from economic shocks, international crises, acts of God, bear markets that reduce the value of all stocks—an endless litany of calamity the Treasury Bill holder dozes through, and for what? A dividend check in the mail that's less than half the income of the T-Bill holder. If the investor accepts the doctrine that earnings are just as good as (and in the face of taxation, better than) dividends, there's still little solace—the Price/Earnings ratio of the Dow Jones 30 Industrials stands at 12.3, the reciprocal of which is almost precisely 8%. Thus for assuming all the risks, trusting the management to optimally deploy retained earnings, and adopting the long-term investment posture which is the only way to ride out the fluctuations of the market, the stock investor receives no more than had he bought a T-Bill. Buy a share of America? Sure... and then let me tellya 'bout this bridge I got.

What's a corporate raider to do? Here's a profligate management, squandering the company's resources on so-called "reinvestments" which, despite their self-evident risk, yield less return than riskless short term government debt. Since corporate raiders are instrumentalities of the Efficient Market and Self-Sacrificing Servants of Society, their actions are merely the means through which a tortured economy seeks equilibrium. Let's see how a company looks to its investors after the raider has struck. Before, the company was as we described above—funded by shareholder's equity and retained earnings, paying a return to the shareholders, in today's

[287] TurboFoam Kleen-Sneak™ is a trademark of Marinchip Systems, so keep your grubby predatory hands off, O.K.?

environment of high interest rates, less than that available from risk-free short-term debt. After the raid is complete, the company's capitalisation has undergone a dramatic change. The previous shareholders' equity has been eliminated; equity is now concentrated in the hands of the raider and his small band of capitalist running dogs. The company has assumed a huge burden of debt—in the purest case of leveraged buy-out the debt equals the entire market capitalisation of the company.

How will that debt be serviced and paid down? By reorienting the corporation from reinvestment in its business to generating cash to repay the holders of its bonds. That this is possible is a product of the tax system and the valuations placed on companies by the stock market. First, the tax gimmick: interest payments on debt are deductible. The same policy that inflated real estate values into the stratosphere is a proximate cause of the takeover boom as well. Consider: if a corporation earns a dollar and pays it out as interest to a bond holder, the creditor receives a full dollar (pre-tax). If the corporation retains the dollar of earnings for its own use, it must pay corporate income tax on it, which shears 33% to 40% from the original dollar. Consequently a corporation can pay out almost 50% more to a bondholder in interest than it could pay a shareholder in dividends, purely as a consequence of the deductibility of interest payments. So, if the corporation uses its entire cash flow to pay the interest on the debt undertaken to buy it out, it can pay the bond holders its before tax profit, 12.3%, a 50% premium over the income to be had from the Treasury bond—enough to pique the interest of even conservative investors.

Second, the acquirer of the taken-over company usually pledges to sell off some of the assets of the original company to retire some of the debt undertaken in the acquisition. That this makes sense is indicative of an inefficiency in the market which has a rational basis in fact. Since, as we've seen, the yield returned by a profitable business to its investors is less than they can obtain without risking their capital at all, the market quite rationally values these investments below their liquidation value to one able to realise all the value inherent in them. If there is another company able to gain market share, earnings on the margin, or other benefits from the acquisition of portions of the original business, it is reasonable to expect that these portions can be sold for more than their beneficial contribution to the sales and earnings of the selling company.

What happens when a leveraged buy-out runs its course? A corporation which previously followed conventional guidelines of reinvestment, dividend payments, and service of modest debt has been transformed into an engine that generates cash flow to service the interest payments due the creditors who financed the acquisition. The result can be viewed as the unbundling of the earnings of the corporation from the possibility of appreciation of its equity—the new owners promise to pay bondholders substantially all the current earnings in the belief that they can restructure the corporation to yield additional earnings which will accrue directly to themselves.

Recapitulation

The current rash of corporate takeovers are the consequence of a historically-unprecedented circumstance: the discount rate substantially exceeding the expected yield from reinvestment of profits in well-managed, growing businesses. This situation, which has resulted in the disappearance of many companies which existed for decades, illustrates the significance of the gap between the after-tax earning potential of a company and the prevailing discount rate.

Theme 2: Leverage vs. debt

Leverage in finance is the control of assets which exceed the direct capital invested in their control. Leverage usually involves the assumption of debt, but leverage is not synonymous with debt. Inherent leverage, leverage without debt, is central to many investments, and is essential to understanding the New Technological Corporation.

Second prelude: A dirt mine in Idaho

Your first reaction is the universal reaction, "*This* is a silver mine?". Having become a shareholder in Bitter Luck Next Time Mining Corporation, traded on the Spokane Stock Exchange, you've taken it upon yourself to pay a visit to your investment. What you appear to have bought for the princely sum of $0.03 per share is... a hole in the ground. A hole in some singularly barren ground.

Yet what you own is a legitimate mining property, one which exhibits great financial leverage without involving a penny of debt. Your penny stock investment can, under the right circumstances, make you a millionaire. Understanding the difference between leverage and debt prepares us to tackle the greatest manifestation of leverage of all: the technological leverage of the New Technological Corporation.

Leverage through debt

Since "leverage" is so often used as a synonym for debt, let's review how debt leverage works in a common financial transaction—buying a house. Suppose you want to buy a house as an investment, and that the house costs $100,000 (this is a hypothetical example, after all), and you can expect to get about $600 per month in rent from the house. Suppose you were fortunate enough to have the full $100,000 in savings and bought the house for cash. If you sold the house one year later for $110,000, you would have realised a total before-tax gain of $17,200 on your investment, the $10,000 appreciation in the value of the property, plus 12 months of rent at $600 per month. Dividing the proceeds by the investment, you would have realised a yield of 17.2% on your $100,000 capital.

Most people don't have $100,000 in the bank and even if they did, they wouldn't want to tie it up in one investment. Suppose, instead, you bought the house by making a $20,000 down payment and borrowed the balance of the purchase price, $80,000, by taking out a mortgage secured by the house. If mortgage rates were 12%, you'd be making payments of about $800 a month, but since those interest payments are tax-deductible you'd be able to cover them from the rental income. When you sold the house at the end of the year for $110,000, you'd end up with a gain of $10,000 (assuming the rent just covered the loan payments), or a gain of 50% on your investment of $20,000. If you'd had $100,000 with which to play the market, you could have bought *five* houses this way and wound up the year with a gain of $50,000 compared to the non-leveraged gain of $17,200.

This is debt leverage, and it works the same in real estate, trading stocks on margin, or taking over companies in leveraged buy-outs. If it's so neat, why doesn't everybody do it? Because leverage is a double edged sword. Debt leverage simply magnifies the effect of changes in price compared to your original investment. If the market moves in your favour it works to your benefit; if the market moves against you, your losses are magnified and may total much more than your original investment. In addition, once you assume debt you are

committed to making the payments on it—if you miss a payment you can lose everything, so you must be very confident of a continuing flow of cash to service the debt.

These aspects of debt leverage have given it a well-deserved bad name. So many economic cataclysms, business failures, and personal bankruptcies have resulted from debt leverage that its enthusiasts tend to be the lucky and its defenders those with short memories.

Inherent leverage

Debt leverage is but one kind of leverage. Let's return to the hole in the windblown soil of Idaho to examine another. Despite appearances, you have not been taken to the cleaners. Beneath the ground is a vein of silver ore reliably assayed as bearing at least two ounces of silver for every share of stock in the "mine". But wait, you say, visions of wealth swirling before your eyes, silver is $6 an ounce! I'd better buy more stock before this gets out!

Alas, as always, there is A Catch. The ore from the Bitter Luck Next Time Mine is a substance one could describe charitably as "low grade ore" or cynically as "high grade dirt". The total cost of extracting silver from the mine, including development, excavation, and refining, works out to about $9 an ounce. Now the stock price begins to make sense: what's the value of a mine which can produce tons of silver while losing $3 on every ounce?

But suppose the price of silver rises. As long as silver sells for much less than $9 an ounce, your shares will be close to worthless. But for every dollar silver rises above $9, your shares represent $2 of real value. If silver should rise to $50, as it did in 1980, and remain there, as it did not in 1980, then each share in your mine would be worth about $82 ($2 \times (50 - 9)$). If you invested 1000 dollars in shares at $0.03 each, your investment would grow to a more than two and half million dollars. This is leverage: leverage without debt. Shares in the mine acquire value only when the price of silver crosses a specific threshold: the price of production. After that point, they track the price in a linear fashion. If you believe that the price of silver will rise at some time in the future, but you don't know when, you can place a bet on that belief by buying shares in a mine with production cost above the current price, sit back, and wait for the price to rise.

This is only one example of inherent leverage. Anybody with a stock option benefits from another. A stock option can never be worth less than zero at its strike price, but its upside gains are unbounded. Stock purchase warrants, rights offerings, options on commodity futures contracts, and convertible debt securities are all financial instruments which exhibit leverage without the assumption of debt.

Technological leverage

Consider Autodesk, Inc. Autodesk was formed with less than $60,000 in capital, yet less than seven years later has a market valuation in excess of $600 million—an appreciation of a *million* percent, based on essentially no capital investment, no physical plant, and no assumption of debt. This kind of performance has the distinct odor of inherent leverage and, if viewed in that light, reveals one of the fundamental properties of the New Technological Corporation.

What is the essential aspect of inherent leverage as exemplified by the Bitter Luck Next Time Mine? It's the possibility of enormous gains to be had if certain future events occur, with no out of pocket costs while waiting,

and the ability to fund development of the assets from operating revenue should success smile upon the venture. What is different about the software business?

Our cost to develop and launch a new product is minuscule compared to the revenue generated from a successful product (assuming we focus on establishing new product categories rather than attempting to "buy market share" in a market dominated by others). The total expenditure involved before the decision whether a product merits further development and promotion or discontinuation is small. The value of becoming the *de facto* standard in a market is enormous. And yet the downside is no more than a write-off of the product development funds—a tiny sum compared to the capital costs of any other business.

The huge difference between the sunk costs of product development and initial marketing and the ongoing revenues from a success constitute leverage just as much as does the production cost of a mine compared to appreciation of its end-product, or the strike price of an option compared to appreciation in the underlying security or commodity. In the case of technological leverage, funds committed to new product development are multiplied by a huge factor in the revenues they return when they yield a successful product. Because the multiplier is so large, the consequences of a failure, or many consecutive failures, are of limited economic consequence except in opportunities foregone to attempt the products which failed.

Technological leverage is the economic consequence of the value of information. Technological leverage translates possession of information into economic value, multiplied by market position and the ability to compound the initial success by delivering follow-up products to the original customers. Inherent leverage always involves a nonlinear price function. In the silver mine case, the cost of production set a floor on the profit curve. A software company with a successful product turns blank magnetic media and paper purchased for pennies into products sold for thousands of dollars by adding nothing but information to them. Technological leverage in the software business stems from the tiny costs of product development as opposed to the enormous ongoing profits of success. The flip side of technological leverage is that possession of large capital resources does not confer a competitive advantage (although the credibility of an established vendor and the access to distribution channels attendant upon that position has value).

To believe in technological leverage is to acknowledge that information has a value equal to or greater than financial assets. Information—embodied in a computer program, a perception of a market niche as yet unexploited, or a new way to organise a business in a market considered saturated by look-alike competitors—is capital. Capital acquired without cash constitutes leverage. Technological leverage is the capital that inheres in information, and therefore is the most powerful leverage of all. The New Technological Corporation is a corporation in which technological leverage is the predominant factor relating product development investment and operating results. The astonishing success of such corporations can best be understood in that light.

Recapitulation: nonlinearity and gain

Debt leverage is linear: it magnifies the gain or loss resulting from an investment of a given size. Inherent leverage is nonlinear but continuous: it exploits nonlinearities in the price/value curve of an investment to produce gains, often without the carrying costs or symmetrical downside risk of debt leverage. Technological leverage is not only nonlinear but is often discontinuous: the introduction of new technologies can cause discrete jumps in the economic fundamentals of a business, an industry, or an entire economy.

The experience of the last two decades of technological innovation, exemplified by Moore's law of semiconductor pricing, the exponential growth of computing power at constant cost, and the manufacturing, product cycle, and investment consequences of the replacement of machinery with software, bear witness to the power

of technological leverage, the rewards that accrue to those who employ it to their benefit, and the risks to those who ignore it.

Debt leverage carries with it the risk of bankruptcy. Technological leverage bears the risk of obsolescence. Those who profit by technological leverage are running on a treadmill whose speed increases as technology advances. To fall behind is to be cast out of the game with little hope of re-entering as the pace continues to accelerate. Unlike debt leverage, technological leverage poses a "keep up or give up" choice to businesses, as the makers of mechanical calculators and watches learned too late.

Theme 3: The talent-constrained enterprise

> *The methods and patterns of growth of a business are often a consequence of the factors that constrain its growth. Most businesses are constrained by capital costs and well-understood product, manufacturing, and market limits which capital can be used to overcome. The growth of a New Technological Corporation is constrained by the supply of talent to create the technologies from which its technological leverage flows. Capital is of limited use in overcoming this constraint.*

Third prelude: What is this, really?

When Autodesk was planning their initial public offering, one key question to be decided was "Is this a CAD company or a software company?". This was not a matter of publicity nor clear communication: millions of dollars depended on the answer. At the time of our offering, personal computer software companies were out of fashion and each dollar of their earnings was valued at about $6.50 in stock price (in other words, the price/earnings ratio, or P/E, was 6.5). CAD companies, however, were the Going Thing, and commanded P/E's of about 13. So, if you were a CAD company making precisely the same number of dollars on the same volume of sales, your stock would be worth twice as much as a software company reporting identical numbers. How to classify a personal computer software company whose only product was in the CAD industry? Easy: look at the numbers and say, "Yessiree—we're a CAD company, all right".

One of the most difficult issues in performing an initial public stock offering is arriving at the valuation of the company—in other words the stock price of the offering. An incorrect valuation can have disastrous consequences: too high and the underwriting syndicate takes a bath on the offering and the lead underwriter may find it hard to fill up the next syndicate; too low and the company selling the stock foregoes millions in proceeds and may take its much more lucrative follow-on offerings somewhere else. For a process that involves more intangible factors than most engineers believe exist in the entire world, it works almost perfectly—if product introduction disasters were as rare as underwriting calamities the world would indeed be "entrepreneur friendly". Assigning a valuation to a business is a subjective matter relying on the judgement of individuals who probably could not begin to explain how they arrive at the numbers they do, but the first and most important determining factor comes many months earlier when management answers the question "what kind of business is this, anyway?". Their answer is what I refer to as the "shape of a business".

The shape of a business

The "shape of a business" is manifested in its form of organisation, capital requirements, methods and patterns of growth, personnel requirements, and the risks to and yield from the capital invested. Businesses in the same industry will tend to have the same shape; significant deviations, unless clearly manifestations of obvious success, usually mean something is wrong within the business. This shouldn't be surprising: businesses evolve within the pattern of competition and cooperation of the marketplace much as organisms evolve within an ecosystem. Just as biology tends to find similar solutions to similar problems from many different starting points, the market tends to drive businesses with the same fundamentals to the same optimal operating ratios.

The shape of a business is often a consequence of the ultimate constraints on growth of the enterprise. Most businesses are constrained by capital costs, material cost and availability, product development cost, manufacturing capacity and costs, and market size and demographic factors. The tradeoffs among these constraints are well understood, and capital can be deployed to ameliorate any of them.

The shape of a business is reflected in the financial aggregates that measure its performance. Within a given industry, operating ratios tend to converge upon the same results. These results, in turn, can be interpreted to identify the key resources on which the growth of the business relies. In the Nineteenth century, one of the epochal events was the building of the web of railroads that interconnected each continent. Constructing a railroad required rights-of-way, largely secured by government concessions through the right of eminent domain, access to large amounts of capital to finance construction and initial operation, and labour where required for construction. Railroads were thus largely a creature of the debt market and government policy, and it was railroads which first introduced the concept of 100 year bonds and, in a few cases, perpetual bonds to the credit markets. Much of what we now call "heavy industry" similarly depended upon debt financing—wherever a massive physical plant had to be constructed before revenues could flow, debt was at the heart of the business.

Each business finds its own shape, and with that shape, the mechanisms for financing its development and growth. The intimate association of private venture capital pools with semiconductor-based high technology is a consequence of the ratio of the start-up capital costs and business development times characteristic of that business compared to the cash-out time and expected yield. Businesses with comparable 10-year risk/reward ratios, such as private satellite launching, new information utilities, and desktop chip fabrication must seek funding through other channels because the shapes of their businesses are incompatible with funding mechanisms which co-evolved with the development of more conventional businesses with which they contend for funding.

The growth of a New Technological Corporation, however, is largely constrained by the availability of talent. It is talent that identifies opportunities created by technological growth, defines and develops products to exploit them, and markets and sells the products to establish them before the niche is occupied by competitors.

Wild Talents

The talents essential to a New Technological Corporation are rare, hard to find, and difficult to identify even in an interview. They are often prone not to repeat even after a stunning success. Charles Fort's term "Wild Talents" may be appropriate to the central asset of a New Technological Corporation. Reliably staffing and expanding those positions that create technological leverage is often as frustrating and seemingly impossible a task as seeking the Holy Grail or attempting to find a repeatable and unambiguous demonstration of parapsychology. Yet it is the presence of such talent and the ability to bring the products it develops to market in a timely fashion that secures the future of a company in an information-intensive industry.

There are few prototypes of talent-constrained industries available for study and the parallels one can find are imprecise and often misleading. The business closest in economic "shape" to the software company may be, to the surprise and dismay of technologically-adept software developers, the advertising agency. An advertising agency can be viewed as an inverted pyramid with extensive account relations, production, purchasing, research, marketing, and management resources which mediate the interaction between the agency's client base and a small pool of creative talent who generate the concepts that drive the campaigns that the agency, as a whole, creates.

Technologists' disdain for this economic parallel does not erase the fact that the Wild Talent that invents messages such as "The Pepsi Generation", "The Heartbeat of America", "The IBM Commitment To Service", or "Tools for the Golden Age of Engineering" creates capital just as surely as the Wild Talent that invents new computer applications or makes existing applications widely accessible at low cost. Both talents create an intangible product: *pure information* which, once released into the market, yields sales and profits thousands of times greater than the cost of creating the idea which yielded the wealth.

Recapitulation

The New Technological Corporation has a unique economic "shape" reflecting its limited capital requirements, low cost of goods, and low cost of product development. Its shape results from the technological leverage created by a small number of "good ideas" which have become accepted by the marketplace.

No company has found a way to successfully generate such ideas on a production line. Ideas flow from intermittently talented individuals who are difficult to attract and retain, and it is also difficult to screen good ideas from bad without testing them in the market.

The shape of a New Technological Corporation derives from the chief constraint on its growth, the ability to generate ideas that create its technological leverage. Financial capital is of limited use in accelerating or increasing the flow of these ideas.

Theme 4: Quantum economics

> *Modern physics tells us that reality resides not in aggregates but in discrete interactions and transformations the aggregates only dimly reflect. Economics, and the understanding of business built upon it, largely relies on interpretation of aggregates with continuous behaviour. These measures may reflect the behaviour of markets in which discrete transactions are the only reality no more than large number aggregates describe the underlying events of physics.*

Fourth prelude: Aggregates aren't reality

In the midst of preparing Autodesk's public offering, I suddenly realised that investment bankers and accountants actually believed there was a causal relationship between the percentage of sales spent on R&D or marketing and the time-delayed sales and profitability of the venture. I had never even calculated such numbers, much less assigned any significance to them, focusing instead on what specifically needed doing, then how much

could be done with the resources at hand. After recovering from the offering, I began to think about the idea that "aggregates aren't reality".

The process of managing a large and growing business is very much a matter of learning how to interpret abstract aggregate measures of the performance of the business and thereby deciding what specific actions to take. In the midst of struggling for survival while learning that skill, it's hard to remember that:

> Aggregates aren't reality
> Reality is events, not a process.
> Reality is discrete, not a continuum.

Economics as events

> "Every heat engineer knows he can design his heat engine reliably and accurately on the foundation of the second law [of thermodynamics]. Run alongside one of the molecules, however, and ask it what it thinks of the second law. It will laugh at us. It never heard of the second law. It does what it wants. All the same, a collection of billions upon billions of such molecules obeys the second law with all the accuracy one could want."
>
> — John Archibald Wheeler[288]

We construct aggregates to approximate the behaviour of large numbers of discrete interactions. Sometimes they are useful, as in thermodynamics. Often they aren't, as with most macroeconometric measures. Wheeler suspects that *all* our laws of physics describe approximate behaviour of aggregates of observations; that the fundamental quantum event is all that really exists. Most of physics does not attempt to understand why these quantum events occur but simply describes the aggregate behaviour of large numbers of events. As we begin to understand the low-level mechanisms, we will get to the true physics beneath the aggregates. Similarly, in economics we try to predict behaviour of aggregates of individual transactions. Only the transactions are real; all the rest is the work of man. One may not be able to understand what drives the transactions by theorising based upon aggregates.

Parallels exist between markets and quantum mechanics. The electron has no position or momentum until you measure it. When you measure its position, you disturb it, foregoing accuracy in measuring the momentum. A share of General Motors has no price until a buyer and seller exchange it, a discrete event. This transaction/measurement affects the price of subsequent transactions. Prices are undefined until a transaction occurs, whether the purchase of a loaf of bread or the takeover of RCA by GE. Prices in a large liquid market can be predicted quite well since the effect of a single transaction is minuscule; prices in blockbuster transactions can barely be predicted at all. Similarly, you can predict interference fringes to many decimal places but which detector an individual electron will trigger in a dual slit interference experiment is unknowable in principle.

"The market was up 15 points today" is meaninglessness layered on meaninglessness. The market is neither up nor down. The market is a *place* where discrete transactions occur—a surging organic sea of buyers and sellers with different goals, opinions, and strategies, who momentarily and unpredictably agree to exchange specific assets. We aggregate these transactions into the abstraction of a continuum of price. We aggregate a selection

[288] "World As System Self-Synthesized By Quantum Networking", *IBM Journal of Research and Development*, Vol. 32, No. 1, January 1988.

of these abstracted continua into an average price. We then assign meanings to the action of this average, and impute its behaviour as being representative of the market.

Thermodynamics works because the number of particles is a statistical universe. Economics may not work because the number of players and events is too small. Perhaps the fundamental difference between people in a market and gas molecules in a jar is not that people have free will and gas molecules don't, but just that there are a lot more gas molecules.[289]

The further you are removed from the events, the less you're able to see what is really going on. MBAs and investment bankers are trained to look only at aggregates: "Well, if they're putting 10% of sales into R&D, that will translate into a 30% sales increase in 2 years" or "Their margins are eroding, and therefore…". Local governments work pretty well because the people who run them are actually aware of the sewers, potholes, and running dogs. Large national governments can deal only with totally abstract aggregates and consequently are less effective. Decision making must, to be effective, be based upon accurate information regarding events. To the extent that government or business managers see and adjust only aggregates, their actions become increasingly ineffectual. As those governed or the customers of a business perceive consistently ineffectual or counterproductive actions, the legitimacy of the institution wanes.[290]

Recapitulation

The fundamental event in a business is the purchase of a product or service by an individual customer. Margins, percentages of sales, sales trends, return and defect rates, and customer satisfaction indices are all abstractions from aggregates of events. They may prove useful diagnostic tools but they are not reality. Understanding why the discrete events occur may be more useful than any of the aggregates.

Collectivism and central planning, whether in government or in the management of a large business enterprise, embody the Nineteenth century view of the world as grand machine. One can design and improve a machine. Classical liberalism is much closer to the Twentieth century interpretation: society as an aggregation of discrete events. At most one can control incentives (as one can affect a thermodynamic system by increasing the temperature or compressing it), but attempting to prescribe events doesn't work any better than Maxwell's demon.[291]

Since the wealth of a New Technological Corporation derives in large part from the technological leverage created by discontinuous shifts in the marketplace caused by a small number of innovations, it is essential that managers of the venture remain in touch with the low-level events that determine the destiny of their company. Turning a knob that controls an aggregate such as increasing research and development spending by 25% or shifting funds from marketing of an existing product to promotion of a new product will not have predictable results. Only by understanding the precise points at which the company's technological leverage

[289] Does this mean that "psychohistory" can emerge as the number of humans in the universe surpasses Avogadro's number? How could that number of participants in a market interact given the volume they would occupy and the speed of light?

[290] Politicians sense this when they campaign in front of a closing factory or bankrupt family farm—they're trying to tie their (aggregate) policies to a (discrete) event. But more and more, even those in the crowd or those affected doubt there is a link between what the politician proposes and events actually changing.

[291] One of most enlightening indicators of how deeply information is embedded in the structure of the universe is the discovery that Maxwell's demon fails not because of inability to measure the momentum of the molecule but rather because of the energy consumed in destroying the information from the last measurement. This seems to indicate a deep relationship between destruction of information and irreversible processes. Might one view attempts to control and prescribe at the transactional level (e.g. minimum wage, price controls) as failing the same way as Maxwell's demon—the nonlinearity at the transaction level destroys information from the market essential in providing the feedback that makes the market function efficiently?

is applied, then carefully analysing the reasons which lead customers to select the company's products (or a competitor's product) can useful strategic decisions be made. This requires that senior management receive accurate, extensive, and unbiased evaluations of the development of technologies related to the company's markets and act promptly to maintain and expand the company's leverage.

Theme 5: Equilibrium and efficient markets

Financial analysts generally assume that markets are "efficient": that prices reflect all the information known to market participants and that consequently the market sets accurate prices for the assets it trades. Market crashes, large shifts in the relative valuations of industry groups, and other fluctuations without apparent causes are difficult to explain in these terms. Perhaps markets are efficient only when near an equilibrium point and cannot be relied upon for accurate feedback in the presence of rapid or discontinuous change.

Fifth prelude: October 19th, 1987

The NASDAQ National Market System on which Autodesk, Inc. stock is traded maintains a market surveillance office to monitor activity in stocks and attempt to detect unusual price changes, unexpected increases in trading volume, or other action which might indicate a stock reacting to information not yet publicly disclosed. When the action of a stock triggers the monitoring computer's filter, a person in the office calls an officer of the company to inquire whether the company knows of any information which might cause the unusual trading pattern and, if so, when it will be disclosed. The patterns the computer watches for are those that indicate apparent inefficiencies in the market such as strong buying of a stock with little concern for price, which could signal accumulation of the stock by an investor who had illicitly obtained information about an impending takeover.

On October 19th, 1987, action in Autodesk's stock tripped the warning and Al Green received a call from NASDAQ's market surveillance office to ask if "there was any reason for the unusual action in Autodesk stock". Let's see, could it be that the call was placed right in the middle of the worst global financial crash in the history of economics? Quite likely....

While humorous, the event limns a deeper unity between the efficiency of a market and its closeness to the point of equilibrium between buyers and sellers. It is well known that a market can be efficient only if it is liquid: that is, has enough transaction volume so buyers and sellers are readily matched. In a "thin" or illiquid market a slight imbalance between buyers and sellers, even if momentary, can cause large swings in price unrelated to any underlying property of the asset being traded. That Autodesk stock exhibited the symptoms of an inefficient market on a day that broke all records for trading volume demonstrates that volume alone does not guarantee efficiency. Efficiency may require that the market be close to an equilibrium point in the physical sense: where not only are buyers and sellers closely matched in numbers, but that they share information, beliefs about the future, and models of valuation which form a continuum with a single modal point.

A quiet, normal day

The trading floor of the Chicago Board of Trade during business hours on a normal day would fit anybody's definition of chaos. Each trading pit is filled with screaming, arm-waving, gesticulating traders jumping up and down, scribbling on little pieces of paper, and handing notes back and forth to "runners" shuttling to and from the wire terminals where orders are received. This is an efficient market at work on a normal day. Since there are a large number of orders to buy and sell at many points around the current price (hence the many pieces of paper in the traders' order books), movements in price will be close to continuous. Since there are a large number of buyers and sellers, including floor-based "scalpers" or "locals" willing to make trades of less than a minute's duration to turn a profit of one tick in price, the market can accept large buy or sell orders without discontinuous price changes (it is unusual in a market this liquid for consecutive transactions to differ in price by more than the minimum increment of quotation, even if the overall price swings in a day are large). That a well-balanced, highly-liquid, efficient market near equilibrium looks like a cockfight where somebody forgot the chickens is evocative of the intellectual tension between the apparent messiness and anarchy of markets and their usually smooth functioning in practice.

What happens when the market diverges from equilibrium? Two days before I wrote these words Ford Motor Company issued a press release to the effect that their researchers had made major progress in developing a catalytic converter for automobiles that required no platinum. This news hit the platinum market, which had been rising strongly for much of the last year, like a sledgehammer. Now the Ford announcement, which simply reported that patents had been granted on a device which would undergo initial tests in 1989, had absolutely no impact on the near-term supply and demand for platinum, for which automotive catalytic converters represent 30% of the world demand and 60% of the U.S. demand. Nonetheless, the announcement caused a huge number of sell orders to hit the platinum market while most participants scrambled to figure out the actual significance of the development.

What did the platinum pit look like after the news arrived? Chaos squared? No, it was dead. Futures markets have daily trading limits, so when all the sell orders hit the market it simply went down the limit and business ceased because there were no buyers at the limit-down price. This was a market out of equilibrium, a market where the disequilibrium caused volume to dry up, and thus the price-setting function of the market temporarily ceased to function. (Although daily trading range limits are unique to U.S. futures markets, the same effect would have obtained in any other market through different means. On the New York Stock Exchange they call it "stock closed by the specialist due to order imbalance". On NASDAQ the broker-dealers simply remove their bids from the system or stop answering their phones.)

The only thing one can predict with certainty in a market is that equilibrium will be re-established at a new price, and trading will resume its chaotic course from that point.

Equilibrium and information

If markets generate accurate price information when close to equilibrium, what is the prerequisite for efficiency? It is the flow of information. As long as the information being processed by the market is information *about* the market (in other words, the balance between buyers and sellers and the prices they bid and ask), the market will act to maintain the equilibrium by adjusting the price. When exogenous information enters the market, whether the elimination of part of the demand for a commodity, supply disruption such as an unexpected freeze of the Florida orange crop, a change in the prospects for a company's earnings such as that caused by a disaster at one of the company's plants, or the launching of a takeover bid at a premium, the market's equilibrium is

disturbed and the market will move chaotically and discontinuously until it finds equilibrium again.

Whenever a market is using incomplete or inaccurate information to arrive at its valuations, the prices it assigns cannot be relied upon as valid. The large shifts in the valuation of industry groups through time may be seen as the market reacting as it obtains and digests information regarding the events and realities of those industries, which may not be visible in the financial aggregates they report.

We have seen how arbitrary is the process of classifying a company within an "industry group" and how capricious the market can be in valuing these groups. To the extent that the market recognises New Technological Corporations at all, it lumps them with "high tech" and values their earnings within that sector. This aggregation may be incorrect. The great majority of "high-technology" companies are capital-intensive businesses in the producer non-durables sector, characterised by short product cycles, heavy research and development investment, rapid obsolescence of capital equipment, and rapid erosion of margins in a highly competitive market. One can dispute the validity of every single one of these assertions for a New Technological Corporation. This suggests that the market will eventually discover that the "shape" of a New Technological Corporation is not only very different from what it considers "high-tech" but is, in fact, virtually unique among companies. As this realisation dawns and its implications for the long-term earnings prospects of the group are worked out, the market can be expected to re-value the stocks of New Technological Corporations based upon their fundamentals. The properties of such businesses suggest that the revaluation will be substantial and upward.

Recapitulation

Markets arrive at prices for the assets they trade by arriving at an equilibrium between buyers and sellers. When the flow of accurate information about the fundamentals of the market fails to reach the market participants, the market diverges from equilibrium and reports inaccurate prices. Only when the information has entered the market and been absorbed by the participants are equilibrium restored and valid prices re-established.

In a market dominated by institutions with a short-term perspective, relying upon industry analysts with an MBA focus on financial aggregates, information about the events within a company or industry group can take a long time to reach the market. Consequently, there may be a long delay between the emergence of the exemplars of a new industry group and the market's recognising them as a group with its own fundamentals and principles of valuation. In addition, the potential of technology to cause discontinuous changes in values through technological leverage is generally not recognised by the market until what has been called "the creative destruction of capital" is well underway.

The market exhibits little evidence of having distinguished the fundamentals of New Technological Corporations from other "high-technology" companies with very different properties. As the managements of this new group of companies communicate their distinguishing properties to the market both by conventional channels of education (such as meetings with securities analysts and industry forums) and by developing their businesses in directions that exploit the advantages they possess, the market can be expected to revalue their companies.

Variation 1: The New Technological Corporation

The New Technological Corporation is precisely what its name implies. Before we examine strategies such a company might adopt to better take advantage of its unique fundamentals, let's pull together the threads that describe why these companies are what they are. They are:

New. . .

New Technological Corporations have a new financial "shape". This shape is the product of their being talent-constrained rather than limited by more usual factors such as the cost and availability of capital. Their combination of very high operating margins, low capital requirements, and the decoupling of capital investment from future economic prospects marks them as unlike most other businesses.

Technological. . .

The New Technological Corporation derives its "shape" from the technological leverage it employs to achieve such high yield from small capital investments. Because the company profits by technological leverage, its future depends upon maintaining that leverage both by avoiding obsolescence and seeking other products in which technological leverage can be exploited. Because technological leverage is the result of exploiting specific ideas from Wild Talents, analysis of the aggregates of such a company without knowledge of the underlying events may reveal little about its prospects. Because the capital requirements to develop technological leverage are low, the possession of large capital resources and cash flow, while conferring stability in hard times, the ability to make acquisitions, and credibility in the market, may not be particularly useful in maintaining its technological leverage.

Corporations. . .

However different, New Technological Corporations coexist in the market with other firms of all kinds. In the securities markets, the stock of a New Technological Corporation may be incorrectly valued because information regarding its financial shape has not reached the market and the company is incorrectly grouped with "high-technology" companies with very different profiles. In the market for its products, the New Technological Corporation may, by failing to understand its own fundamentals at the event level, forgo competitive advantages unique to it when competing against companies with different profiles.[292]

It is therefore in the interest of a New Technological Corporation to understand what distinguishes it from other companies, to exploit the advantages and palliate the penalties those distinctions confer, and, in the belief that the securities market miscomprehends and undervalues New Technological Corporations to explain, by word and deed, these distinctions to market participants.

Variation 2: What to do with the money?

The maturing New Technological Corporation faces a challenge almost unique in the annals of legitimate business: deciding how to dispose of the large and growing stream of earnings generated by its successful products. The fundamentals of its business make the happy circumstance of high earnings an occasion for making some difficult choices.

[292] A competitor may, for example, be forced into massive capital commitments to upgrade hardware to meet competitive pressure created by a small programming change in a product of a New Technological Corporation, thereby assuming a large debt burden in response to advantages obtained by a New Technological Corporation through technological leverage

Reinvest it in the business?

As the standard prospectus language goes, "The Company currently intends to retain earnings for use in its business. . . ". Fine, but precisely how? It's when faced with answering this question that the chief executive of a New Technological Corporation begins scribbling notes for his book, "Technological Leverage—Problem Or Curse". Once a company has amassed a pool of capital adequate to ride out any conceivable financial cataclysm and respond to a competitive assault by any of the players who might challenge the company's position in the market; is promoting its current and emerging products and developing and maintaining its product line from current cash flow; is paying corporate taxes in the highest bracket; and is still generating piles of cash, this question becomes not just "a problem it's nice to have" but one that demands an answer.

The realities of technological leverage and the prevailing cost of money make the answer hard to arrive at. A New Technological Corporation seems to be what every investor dreams of in times of high interest rates: a business whose return is comparable to the debt securities that contend with equities for the investor's cash. Unlike the takeover target whose management cannot reinvest earnings with an expected yield competitive with riskless Treasury Bills, and must be compelled to return the earnings to their shareholders by the reality or threat of a leveraged buy-out, the management of a New Technological Corporation faces a different dilemma: the earnings of their corporation are exemplary and yet they cannot reinvest them at comparable yield, not because yields in the company's business are below those of the debt market, but because throwing money at Research and Development is like pushing a rope; it does not reliably generate the ideas and products from which technological leverage and future revenues flow.

Since the company's earnings come from the unpredictable results of Wild Talents, the company should obviously take every step possible to attract, retain, motivate, support, and efficiently translate the yield of its talent resources into products. But while that process may seem wasteful, inefficient, and indulgent of spoiled eccentrics, the business reality is that it doesn't cost very much compared to the earnings of a successful New Technological Corporation, so taking this obvious step (though neglected by managements that fail to understand their New Technological Corporations) does not materially affect the deployment of the earnings of the enterprise.

Retain it and grin?

Whether by conscious strategy or default, most New Technological Corporations have adopted the strategy that requires no action: simply paying taxes on the earnings and investing them in short-term money market instruments (high-finance for "putting them in the bank"). This strategy makes a tremendous amount of sense, up to a point, that point being, to adopt a cynical turn of phrase, "as long as you can get away with it".

If you believe that New Technological Corporations are undervalued by being grouped with capital-intensive "high-technology" companies, then you may be inclined to excuse the Great New Technological Corporation Price/Earnings Scam as a rational response to a market that refuses to see through the aggregates to the reality of their business. For what happens when a New Technological Corporation accumulates a large pool of financial assets is so remarkable and contraindicative of the concept of an "efficient market" that it's amazing it's still legal.

A closed-end bond fund (or unit trust) is a financial vehicle that collects money from a large number of individuals and uses the sum to purchase a diversified portfolio of bonds with given criteria of quality, composition, and maturity. Each investor owns a percentage of the total portfolio and benefits from diversification among

companies and industries and economies of scale he would not have been able to take advantage of had he bought the securities directly. A closed-end bond fund is easy to value: one simply takes the total income of the fund and the market value of the securities it holds and divides by the number of shares held by investors to establish the yield and price per share.

Consider, now, the New Technological Corporation in its guise as a covert closed-end bond fund. Since the market has not yet distinguished the New Technological Corporation from high-technology corporations, it is far from realising that a significant fraction of the earnings of a New Technological Corporation come from its holdings of short-term fixed-income debt instruments. Consequently, the contribution to earnings from the company's financial assets are multiplied by the price/earnings ratio appropriate to a high technology company and so reflected in the stock price. At this writing, Autodesk, Inc. trades at a price/earnings ratio of 22, and short term interest rates of about 8% translate into a price/earnings ratio of 12.5 for short term debt. Therefore, each dollar Autodesk earns from its retained financial assets is valued 1.75 times higher than the same dollar earned by a closed-end bond fund. If this doesn't justify the word "scam", you must at least concede that it's an awfully kind compensation to bestow upon New Technological Corporations in recognition of the difficulties they face.

Pay dividends?

The most conventional course for a company generating earnings above those needed for reinvestment in the business is to simply pay them out to the shareholders in the form of dividends: as Midnight Oil puts it, "It belongs to them—let's give it back". Tax policy in the United States, combined with a tradition of high-technology companies not declaring dividends, has made dividend payments unusual among small, high-growth companies. The financial situation of a New Technological Corporation warrants revisiting whether dividends should play a role in the disposition of its earnings.

The issues involved in dividend payment draw on all of the Themes introduced above, plus tax policy, the current and expected state of the economy, the composition of the company's investor population, the relationship of founders to the company, and many of these matters interact in difficult-to-understand ways. The following discussion of dividend strategy is unavoidably lengthy and involved. Its relative length compared to the treatment of other potential dispositions of earnings should not be taken as an endorsement of adopting a dividend policy. Instead, it indicates how complicated the decision to pay dividends may be.

Why not dividends?

Why on Earth should an investor object to receiving income from his stock? Let's review why dividends have fallen out of favour. First and foremost is the notorious "double taxation of dividends", a fixture of United States tax policy for decades. Dividends paid by a corporation to its shareholders are not deductible from the company's corporate income tax, whereas interest payments to bondholders are fully deductible. Dividends constitute taxable income for the recipient, so the original corporate earnings are taxed twice: first at the corporate tax rate before the dividend is paid, then again at the shareholder's tax rate.[293]

Let's consider the ultimate disposition of a dollar of sales collected by a company. We'll assume the company pays a marginal tax rate (federal plus state) of 40% and that the investor holding the company's stock or

[293]There is a special gimmick that reduces double taxation of preferred stock dividends paid to corporations, but that isn't applicable to the dividends on common stock we're discussing here.

bonds is an individual also taxed at a 40% marginal rate. If the company takes the dollar and reinvests it in the business by spending it, for example, on an expense item such as payroll or rent, the entire dollar is deductible and hence is applied to the benefit of the company. Of course the dollar, by being spent, is no longer a dollar of earnings reported by the company; publicly held companies expected to report rising earnings and stable margins must balance spending additional dollars against the earnings expectations of the market. Increasing spending also assumes that the expenditures will increase the value of the company. As we have seen, increased spending does not contribute to the position of a New Technological Corporation as reliably as for more capital-intensive businesses, except if the company can obtain a better market position by increasing marketing and sales expenditures.

Is interest more interesting?

If the company has assumed a significant debt burden, the dollar can be applied to debt service (interest payments). Since interest payments are deductible, the company pays no corporate income tax on the dollar, which flows directly to the bondholder. The bondholder must pay tax on the interest he receives, and ends up with $0.60 after tax. Since interest payments are an expense, funds used to meet them are not part of the company's earnings so all the considerations about earnings expectations apply to interest payments as well. While assuming debt is an efficient way to transfer company revenues to holders of its securities and, as we have seen in the case of Leveraged Buy Outs, are used explicitly to that end to rectify the situation where a company cannot reinvest earnings at debt market yields or better, in general debt only makes sense in cases where one needs the capital borrowed. In a business with little need for capital, taking on debt makes no business sense except as part of a takeover defence or subterfuge to return pre-tax earnings. It is unlikely in the extreme that the securities market would welcome a large junk bond offering from a cash-rich, non-capital-intensive business with no need for the proceeds of the offering. In addition, debt carries with it all the risks of debt leverage, foremost among them the risk of bankruptcy in the event of inability to service the debt. Since this negates the key strong point of a New Technological Corporation, its technological leverage without debt, it would seem a highly unadvisable course.

If the sales dollar is neither spent on the operations of the business nor paid in interest, it becomes a dollar of pre-tax earnings. First in line, of course, is the tax man, who lops off his 40% for the Common Good. The remaining 60¢ becomes after-tax earnings, reported to the shareholders in the next operating statement. If the company simply retains the income and invests it in money-market instruments, it simply adds to the company's cash pile which is the beneficial property of the shareholders. Earnings from the cash hoard are, as noted above, aggregated with earnings from operations and may, if things don't get too far out of line and nobody notices what is happening, be reflected in the stock price at a P/E befitting a high technology company rather than a Treasury Bill.

The cost of double taxation

If the company chooses to pay out the earnings as a dividend, the shareholder gets a check for 60¢, the earnings that remain after corporate income tax. The dividend check being taxable income, the shareholder must pay 24¢ (40% of 60¢) of tax on the dividend, leaving 36¢ of the original dollar earned by the company. Although these numbers will vary depending upon the tax rates paid by the company and the investor, it's clear that with taxes taking 64% of every dollar, operating a company in order to pay revenues out as dividends is a far more effective way of transferring wealth to the government, which ends up with 64¢ from each dollar, than to the shareholder, who's left with 36¢.

Viewed in this light, even to contemplate paying dividends may seem the purest lunacy. There are, however, a few more facts to consider. If a business can neither spend its earnings productively (or must generate after-tax earnings to satisfy market expectations), nor has a need for debt which would transfer before-tax earnings to bondholders, payment of corporate income tax is unavoidable. Once the earnings have been booked only two alternatives remain: add them to the company's working capital pool or pay them out. Once the company has amassed working capital adequate for its needs, the shareholders begin to become restive. They demand, and rightfully so, "If you can't think of anything to do with the money other than buy Treasury Bills, why don't you give it back and let us decide how to invest it?". After all, once earnings are reported, payment of corporate income tax is a foregone conclusion. The shareholder does not look at the fraction of pre-tax earnings retained; he sees the after-tax earnings per share reported by the company, multiplies that by his holdings, and begins to think how nice it would be to find a check for that sum in his mailbox, notwithstanding the need to pay taxes on it.

Dividends as an equaliser

Mature companies in stable businesses pay dividends because they have become entities whose purpose is generating earnings for their shareholders. Utilities provide the purest examples of such companies. A shareholder in Pacific Gas & Electric, for example, currently receives an 8% return on his investment in PG&E stock. Why buy a stock that yields less than a Treasury Bill (and on whose dividends you have to pay state tax, unlike a Treasury Bill)? Because the stock can be expected to grow as the demand for electricity in California grows. While collecting income comparable to the T-Bill, you stand to profit from an investment likely to grow at a rate comparable to that of the economy of California, historically a pretty good bet.

Now that you're thinking in terms of balancing immediate rewards in the form of dividends and deferred capital gains from appreciation of stock if a company is successful, several other strategies seek your attention. Over there is a guy in a blue suit hawking IBM stock, "You can take home a yield of 3.6% off the top, and buy in to the most successful stock in history, with a record of 15% compounded sales and earnings growth…". A fella in a plaid jacket and yellow shoes screams, "People gotta eat! General Mills will pay you 3.7% and deliver growth as reliable as breakfast". In the back of the room, behind the nickel slots, are disheveled mute characters wearing signs around their necks. "Who are they?", you ask. They are the stocks that pay no dividends, but each sign ends with the phrase "*huge* capital gains, *real soon* now".

Dividends can be seen as equalising the valuation of companies at different stages of maturity. Ford Motor Company cannot possibly promise you sales and earnings growth, starting from its share of a mature market, equal to that of Digital Datawhack, but it can pay you a solid 4.7% on your money while promising serious capital gains and dividend increases if Ford products gain increasing market share.

To decide whether dividends make sense for a New Technological Corporation and if so, at what level, we must decide where the shape of its business places it on the industry maturity curve: the key determinant of dividend policy when dividends are viewed as leveling the risk-reward tradeoff among stocks by paying earnings to an investor in equities with less potential for capital appreciation. However, since tax policy is so intertwined with the decision to pay dividends, we must first examine two additional tax considerations.

Tax-exempt investors: the privileged many

Since the first tax bite was taken from the dollar of sales at the point the company decided to report it as earnings instead of spending it, the only tax that affects the decision whether to retain after-tax earnings or pay dividends is the tax paid by the recipient of the dividend. The majority of the stock of most high technology companies and, by their inclusion in that group, New Technological Corporations, is held by institutions. Many of these institutions pay no taxes either because they are tax-exempt, as are most pension funds, or by virtue of returning all earnings beneficially to their shareholders, as do most mutual funds. An institutional tax-exempt shareholder in a company with high earnings may view dividend payments in a very different light than an individual investor. The professional fund manager who invests in a company is basically paid to return yields greater than those achievable from Treasury Bills. If a company he invests in cannot think of anything more productive do with its earnings than buy Treasury Bills, he has every right and reason to insist that profits be returned to him for investment at the higher yields his investors hired him to obtain. In addition, whether managing a diversified fund or a narrow industry-indexed fund, the portfolio manager desires a "pure play" in the main business of the companies he selects for his portfolio. It's not clear where a "combined personal computer software manufacturer and money market fund" fits into the picture.[294]

Capricious Congress

Tax policy is not a constant factor investors can include in their calculations. The 1980's have seen dramatic shifts in the tax system. Each change has shifted the marginal rewards of various investment strategies and has thereby engendered a redeployment of assets into those instruments with the greatest after-tax yield. One of the largest items on the policy-making agenda at this writing is changes, probably in the tax system, to come to terms with the "takeover binge". This can take many forms; two obvious approaches are foremost. If anti-takeover legislation attempts to limit the deductibility of interest on debt issued to finance acquisitions, there will be little impact on the issue of earnings allocation by a New Technological Corporation. If, however, the disparity between return of corporate earnings to shareholders and bondholders is addressed by measures that eliminate or substantially reduce the double taxation of dividends, the disincentives to dividend payments will be removed and the market will, in all probability, assign a greater value to dividends which will be reflected in appreciation of stocks which pay dividends.

It behooves the management of a New Technological Corporation whose secure earnings could easily sustain substantial dividend payments to monitor changes in policy which affect the economic incentives governing dividends and adjust the strategy of their companies accordingly.

The dividend treadmill

Dividends are "declared" when a company examines its earnings and decides how much to pay out as dividends. Regardless of whether dividends are called "regular" or "special", whether they are declared quarterly, semi-annually, or annually, in fact the sum paid is totally at the discretion of the management and directors of the corporation. This makes dividends much more attractive than interest payments to a management worried about

[294]The elimination of "pure plays" and the consequent inability to discern the expectations for investments from economic forecasts may be at the heart of the undervaluation of closed-end equity funds and conglomerates, and therefore the phenomenon of the liquidation value of a conglomerate exceeding its composite stock valuation. The contribution of this factor to the 1980's takeover boom may reward scrutiny.

hard times: you can stop paying dividends whenever you need to, but if you miss an interest payment on a bond, you're bankrupt.

Once a company has adopted a policy of regular dividend payments, however, the expectations of the market set limits on management's theoretically complete discretion to set dividends. If an investor has purchased stock in the expectation of receiving $500 a year in income and one fine day the company announces that it's cutting the dividend in half because it needs to retain the cash to build a new Airship Foundry, the investor is not going to be pleased. Suddenly his income has been halved, and the company is going to spend the money on something that may not return value to him for several years, if ever. His natural reaction is to sell the stock and do something else with the money. When many people do this at the same time, the price of the stock gets clobbered and it may take years to recover. Not only has the stock returned unreliable earnings, it has marked itself as prone to capricious changes in dividends, so investors are unlikely to pay as much for whatever income it provides as they'll pay for income from companies which have never suspended or cut their dividends (and there are companies whose record for increasing dividends extends over a century).

Thus, by paying dividends a company creates the expectation that the dividends will continue to flow. Management places itself on a dividend treadmill where failure to meet expectations will result in a sharp fall in the company's stock price. If future earnings cannot sustain the dividend and the company is forced to skip or reduce the payment, the stock will be triply hammered: first in reaction to the earnings themselves, then by disappointing investors who had expected the dividend payment, and finally by establishing a record for unreliable payment of dividends.

In the case of a New Technological Corporation, at least as long as it is grouped with "high-tech" companies, it is not clear that the additional risk to the stock price from missing a dividend is a serious problem. High technology companies merit very high price/earnings ratios based on expectations of rapid and reliable quarterly growth in sales and earnings. The penalties exacted in stock devaluation when a high technology company "disappoints the market" by earning less than the analysts expected are so large that the additional consequences of reducing or eliminating a dividend may not be significant.

The operating margins of a New Technological Corporation are so high that it can sustain a major drop in sales and still generate enough earnings to meet a dividend payment, simply by choosing to pay a larger percentage of earnings as dividends during the sales slump (since the company has no obvious way to reinvest retained earnings, why not meet the dividend?). Also, since a New Technological Corporation is not capital-intensive, the exigencies of its business are unlikely to require retaining earnings for capital spending projects as often happens in high-technology businesses (for example, a semiconductor manufacturer may need to construct an expensive new fabrication plant to remain competitive in its central market). In fact, a reliable dividend payment which results in rising yield as the company's stock declines due to disappointing sales or earnings, or simply because the overall market is declining, can act to moderate stock price swings, as investors who might otherwise sell choose to hold the stock, collect the dividends, and wait for better times. Further, as the stock declines it becomes more attractive to income-oriented investors whose purchases act to stem price erosion resulting from sales by those investing for capital gains.

Dividends and founder-ownership

Because New Technological Corporations tend to be built around a single (or small number of) fundamental ideas and since the technological leverage of this idea allowed the company to grow without large infusions of capital which would dilute the ownership interests of the founders and early equity investors, a New Technological

Corporation is far more likely than most companies to retain, at maturity, a significant ownership percentage by founders.

Founders of such a company will have seen their original investment multiplied thousands to millions of times; they will have attained substantial wealth through appreciation of their original stock holdings. However, as one founder of Autodesk puts it, "They don't take stock at Burger King". So in order to diversify holdings to prevent all of one's wealth being concentrated in a single company—even to see any cash at all from appreciated stock, one must sell stock on the open market. Clearly, any sane founder can be expected to sell some portion of his stock to achieve diversification he can sleep with, but after that point founders often find themselves faced with balancing the desire to retain most of their stock holdings, both to continue to exert influence on the destiny of the company and because they believe the stock a superb long-term investment, and the inclination to sell a portion of their holdings and put the proceeds into income-generating securities.

It is clearly in the interest of any business for founders to retain a significant ownership position. Not only does the company benefit from having a substantial portion of its stock owned by people with an intimate understanding of the company's history and strategy, the founders' stock, being unlikely to be sold capriciously or tendered in a hostile takeover, provides price stability and gives management more freedom to act in the best interests of the company than it would have were all the stock in the hands of institutional investors concerned only with the next quarter's earnings. In a New Technological Corporation where the founders may include some of the Wild Talents whose efforts led to the success of the company, the rationale for maintaining their close involvement is even more obvious.

Adopting a policy of regular dividend payments can significantly reduce the founders' dilemma regarding their stock holdings. Even a modest dividend can generate annual income for founders comparable to the proceeds from the sale of the fraction of their holdings typically liquidated in a year by founders, and much greater than the income yielded by investing those proceeds. Dividends create an incentive for founders to retain their holdings in the belief that the company's future will result in their continued appreciation, without thereby foregoing current income from the capital invested in the venture.

Will income be king again?

Fashions in investments change with time. For most of history, income was the major rationale of investing. Some observers of the economic scene suggest we may be entering a period where the recent fascination with capital gains, inflation hedges, and leveraged speculations will give way to a renewed interest in instruments which generate reliable and substantial income. No economic logic is foolproof, and even the most persuasive argument can be negated by tomorrow's change in tax policy or next week's stock market crash, but the possibility of a general change in the valuation of income is worth considering. How might this happen?

First, the general trend in interest rates has been down ever since they hit historic highs in 1980. If rates continue to fall, as many believe they will, rates on debt instruments may approach and possibly fall below, yields on dividend-bearing stocks (as they have been for most of economic history). If dividend paying stocks become the highest yielding investments, they will become the focus of those seeking income.

Second, tax reform has eliminated the preferential treatment accorded capital gains (in other words, appreciation of stock) compared to income earned from interest or dividends. Investors who previously sought ways to avoid income and realise deferred capital gains to reduce the tax bite no longer have any reason to do so. (Of course, this may change, and proposals to restore the preferential treatment of capital gains are in the air at this writing.)

Third, ours is the age of debt. Debt is growing exponentially, and takeovers are in many cases eliminating equity and replacing it with debt. As the debt market further dwarfs the equity market, debt—income producing investments—becomes the centrepiece of the investment world.

Fourth, there are many reasons to believe that a severe recession is in the offing. A recession and the bear market for equities which usually attends it causes severe depreciation in the value of equities. In such a period, secure and stable income assumes greater value than capital gains, since most capital assets are falling in value.

To the extent that these factors are significant, and the assertion they will increase the relative value of income is valid, the argument for paying dividends is strengthened. Confirmation of these trends would be indicated by a relative increase in value of dividend-paying stocks over comparable stocks which retain earnings.

Mature before its years?

Dividends are usually associated with "mature" companies, whatever that means. What does "mature" mean anyway, and what might constitute maturity for a New Technological Corporation?

Most companies pass through a struggling start-up phase, a period of rapid growth, and an extended maturity characterised by relatively stable sales and earnings. This life cycle usually follows the development of the industry in which the company operates: from not being recognised at all, through exponential growth in a market with unknown total size, to saturation and growth thereafter at rates limited by the overall growth of the market (usually constrained by demographic or economic factors) and the company's share of that market, won or lost at the expense of its competitors. Earnings performance also evolves through these phases: during start-up the company loses money, its losses funded by the original investors. If it succeeds and begins to grow rapidly, it becomes profitable but reinvests all of its earnings in the business to fund its rapid growth and not forfeit portions of the market to competitors who are also growing rapidly. In the third phase the company cannot grow measurably faster by reinvesting its earnings, so it often chooses to pay dividends to its shareholders.

A New Technological Corporation can be expected to follow this pattern of development, but the presence of technological leverage results in a very different earnings profile as it moves from stage to stage. After surviving the start-up phase, a New Technological Corporation begins to generate earnings at a very high rate of return. Because little capital investment is needed during its period of rapid growth, there is little need to reinvest earnings and they are simply retained. After the company's product reaches market saturation, earnings may actually decline as the percentage of sales the company devotes to sales and marketing increases to maintain and expand its market share.

Autodesk's start-up phase ran from April of 1982 through January of 1983, when positive cash flow was achieved. Autodesk is still in the rapid growth phase and, characteristic of that phase, cannot predict when saturation will occur.[295] If AutoCAD reaches saturation and Autodesk does not by that time have another product in the rapid growth phase, Autodesk's revenues will thereafter grow at about the rate of the CAD industry as a whole, between 20% and 35% per year.

Since New Technological Corporations generate earnings during their rapid growth phase which equal or exceed those of mature conventional companies and have little need to reinvest them, one might say that a New Technological Corporation matures early. Its financial maturity is perhaps defined best by having retained all

[295] An industry analyst predicted in 1983 that we would saturate the market at around 12,000 units. We have shipped more than ten times that number to date, with no indication of saturation.

the working capital it needs as an adversity hedge rather than by having saturated its market. This early maturity may justify payment of dividends earlier in the company's life cycle than would be appropriate for conventional companies.

Dividends: a complicated choice

As should now be clear, the issues involved in dividend policy are as complicated as they are profound. What is essential is that the management of a New Technological Corporation reach its decision regarding dividends, whether to pay them or not, with a firm understanding of how the differences between their New Technological Corporation and other companies affect the economic fundamentals and strategic consequences of their decision.

A strategic partnership?

One obvious approach for a company plagued by excessive near-term earnings and a dearth of reinvestment options is to harness itself, by merger with, acquisition of, or substantial investment in a company with a complementary "shape": an enterprise with substantial near-term capital investment requirements and out-year payoff substantially greater than compounded money market returns on the earnings of the New Technological Corporation. Unfortunately, this approach does not seem workable.

First, in an era where short-term interest rates exceed the earnings of mature industry-leading companies, the only investments with the potential to materially better those yields bear high risks to the capital invested. If the New Technological Corporation invests its earnings in such ventures, it risks the wrath of its shareholders for "starting a venture capital fund with their earnings" rather than paying them out as dividends. To the extent that its investments succeed, it dilutes the "pure play" aspect of its stock and becomes instead a composite investment which experience indicates will be valued by the market at less than the sum of the assets that compose it. Finally, there is no reason to believe that the managers of a New Technological Corporation will succeed in identifying promising ventures in which to invest—after all, they readily acknowledge they cannot even reliably predict which products of their own company will succeed.

Therefore, however attractive hypothetical composite balance sheets may appear, partnership with a capital-intense business appears a strategy which will cause vilification of management and shareholder unrest if attempted, collapse of the New Technological Corporation's unique advantages and stock price if it fails, and undervaluation and consequent vulnerability to takeover and break-up if it succeeds.

Make acquisitions?

Is there a rationale for consolidation among New Technological Corporations? In other words, should a New Technological Corporation attempt to grow by acquisition of other companies? Since a New Technological Corporation accumulates a large pool of cash and since its stock bears a high price/earnings multiple, it has the financial muscle to go on the acquisitions trail. Let's see if that strategy makes financial sense and, if so, what kinds of acquisitions should be on the shopping list.

Buying technology

Since a New Technological Corporation exists as a result of technological leverage, the most obvious thing for it to buy is more technology. Buying technology through acquisitions means looking for products in their development or early marketing phases which can be acquired, complete with the Wild Talents who developed them, and incorporated into the company's product line. Since acquisitions at this stage in a product's history tend to be relatively inexpensive, the decisions involved in making such an acquisition tend to focus on how well the product and people fit with the acquiring company, evaluation of the quality and potential of the product, and a buy versus make calculation of the fairness of the price.

If technology and products can be purchased at a price comparable to in-house development or the price premium paid for them is justified by time saved in getting to market, such acquisitions clearly make sense. Most acquisition activities will fall into this category, but since the absolute sums involved are modest, these transactions will have little impact on the overall financial structure of the company.

Consolidation among New Technological Corporations

Rather than acquire an incomplete product or one with little market testing, a New Technological Corporation may look at its cash hoard and market capitalisation and go looking for companies like itself, already on their growth curve, but small enough to acquire and digest. Conversely, one day management may awake to discover that they are being approached with an acquisition offer from a New Technological Corporation senior to them in the financial world. Do such deals make sense? What happens when you put two New Technological Corporations together? You get... a bigger one. Since all New Technological Corporations will tend to have the same shape, the numbers are likely to be proportionately the same, and therefore it is unlikely that the shape of one company will differ much from another.

To determine whether an acquisition makes sense, then, the companies must look beyond the aggregates to the events. If one company brings the other access to distribution (such as a network of skilled dealers, local sales offices, or a major account sales force), technology applicable to the other company's product line (for example, a personal computer database company buying a company specialising in micro to mainframe links), or market dominance in another niche (a PC word processing leader buying the maker of the most popular Macintosh word processing program), then the acquisition can be evaluated simply by studying the technological leverage of the combined companies.

Because New Technological Corporations depend so heavily upon Wild Talents, successful consolidations among them will tend to be friendly mergers. A hostile takeover that results in loss of the Wild Talents responsible for the success of the takeover target will very likely be a Pyrrhic victory for the acquirer.

Acquisitions of conventional companies

Should a New Technological Corporation use its high multiple stock to diversify its industry position by buying companies with other shapes in different industries? It certainly can; few companies have the financial power of a New Technological Corporation in an acquisition. But should it? Probably not. As discussed in terms of "strategic partnerships" above, when a New Technological Corporation consolidates its results with a company with a different shape the sum is almost always less attractive financially. If the acquired company had equal or better margins or capital structure, it would be a New Technological Corporation, not something else.

Managements of New Technological Corporations are, however, well justified in looking over their shoulders at frequent intervals to see if conventional companies are beginning to regard them with envious eyes and slowly, surely, drawing plans against them. As the properties of New Technological Corporations become increasingly apparent, they may come to be regarded as the most attractive of all potential takeover targets. Their liquid assets are enough to pay for a significant part of the acquisition; their large cash flow can cover a large debt load, and their minimal capital equipment and physical plant permits easy integration into another organisation.

The disadvantages in acquiring a New Technological Corporation lie in the premium price one pays for its earnings and its dependence on Wild Talents who can pocket the acquirer's cash and walk out the door if not treated well. These factors suggest that hostile takeovers of New Technological Corporations are unlikely or, at least, unwise.

Variation 3: Competitive strategies

> "You may not be interested in strategy, but strategy is interested in you."
>
> — Trotsky

The unusual fundamentals of a New Technological Corporation suggest several ways in which it can turn its unique attributes into advantages when competing with conventional companies. The following sections briefly sketch some competitive moves that exploit a New Technological Corporation's strengths.

Spending information, not cash

Since a New Technological Corporation's products acquire high retail value by the addition of information to inexpensive raw materials, whenever finished goods can be exchanged at retail price (or even at significant discounts from it) for products and services of other companies, the New Technological Corporation can spend technological leverage as if it were cash: in other words, it can print money.

Incentive programs, contests, exchanges for hardware, co-operative advertising programs: any way at all to use product rather than cash places the New Technological Corporation at a tremendous advantage over any conventional competitor whose cost of goods (most likely involving hardware) is much higher and who derives much less gain from such transactions and may not be able to afford them at all on the scale undertaken by New Technological Corporations.

Cheap development: expensive reaction

The rapid product development cycle and low cost to market of the New Technological Corporation may be turned against conventional competitors who can be forced to spend proportionally far more of their resources to respond. Even though only a fraction of the product introductions by a New Technological Corporation may be ultimately successful, if responding to them consumes resources that competitors might otherwise have spent on effective head-to-head competition, the company may still benefit substantially.

Conversely, nimble and inexpensive reaction can help a New Technological Corporation negate or minimise

the impact of product introductions which cost a hardware-dependent competitor much more time and capital to deploy, and upon which, therefore, the competitor is much more dependent for survival in the marketplace.

Low margin product introductions

By exploiting its minuscule cost of goods, a New Technological Corporation can introduce new products at extremely low prices and still generate substantial earnings during their start-up phases. These low launch prices either force competitors to lose money attempting to respond at a similar price or deter them from entering the market at all, leaving the New Technological Corporation free to move the product up-market by adding functionality at additional cost as the product establishes itself as the standard in the market.

Riding out hard times

A strong cash position and freedom from debt allow a New Technological Corporation to ride out a recession, or even a depression, that creates severe hardships for competitors who operate on much thinner margins. By being able to afford a long-term view, the New Technological Corporation can use hard times to position itself for leadership in the next expansionary phase by continuing R&D and product development while most competitors retrench, by keeping its team together while adversaries are devastated by lay-offs (and recruiting the best people they lay off), and by bottom-feeding for complementary acquisition bargains when nobody else has the cash to buy and everybody else needs to sell to raise cash.

Dividends and the strategy of denial

Adopting an aggressive dividend policy may actually result in denying competitors access to capital. If a New Technological Corporation, by paying dividends, causes relative revaluation of its stock among its industry group peers, conventional competitors whose earnings cannot sustain comparable dividends will undergo relative depreciation. The New Technological Corporation can then use its more valuable stock to acquire technologies, bestowing additional leverage on itself, more cheaply than can its competitors, since their stock is valued less by the market.

Conclusions and recommendations

The decisions involved in running any business are complex enough and have such profound consequences for employees, investors, and customers that to make cut and dried prescriptions is glib at best and irresponsible and destructive at worst. In the process of thinking about the issues that face Autodesk, and in discussions about specific decisions we have made and must make, I have come to the conclusion that I proceed from principles and assumptions about the nature of our business and company that are unusual and at variance with the consensus view of the software industry.

Some of these principles date back to the organisation of the company and before: the idea of developing multiple products and test-marketing, and the effrontery of attempting to start a company with virtually no financial capital were reflections of my belief in technological leverage, although I didn't call it that until last

week. My concern with details, technological opportunities, and bottlenecks stems from belief in what I now refer to as "quantum economics". What I present here is as close an approximation as I can put on paper to the way I think about the issues that affect Autodesk. If the paper is complicated, it is because the issues interact with one another in subtle ways. If the paper seems repetitive, it is because it isn't enough to read about these issues and nod agreement or disagreement: you have to be able to pick them up, turn them around in your mind, see how they fit together, and comprehend how other matters interact with them. In writing this paper I have clarified and made explicit many beliefs I had employed intuitively before. I hope I can transmit enough of the principles I use to think about Autodesk's options, opportunities, and strategies that you can share my conclusions, dispute them on the grounds I used to arrive at them, or reject them with an understanding of the flaws in my reasoning.

Strategy is a lonely business; you never know enough to be confident about any decision and you never know if you're right until it's too late to change your mind. To plot any strategy, you must first know the terrain. If you accept the concept of the New Technological Corporation, then the first thing its management should realise is that they're running one. Proceeding from that realisation, and the fact that their company is therefore very different from most of the companies it competes with and from most stocks considered comparable by analysts, management can begin to examine specific decisions and strategic choices to make the most of the unique advantages conferred by being a member of a new generation of companies: perhaps the first to realise that fact and act to exploit it.

If the management of a New Technological Corporation, fully aware of its financial and competitive strengths, deploys those advantages to the company's benefit, their enterprise will in all likelihood not just be unique, but uniquely successful in the long term.

Generic Software Acquisition

On March 7, 1989, Autodesk announced the acquisition of Generic Software and the integration of the Generic CADD product line into the Autodesk product family. This was the almost inevitable consequence of Autodesk's failure to promote AutoSketch in the U.S. market (in Europe, where AutoSketch was taken seriously, Generic CADD failed to gain the market share leadership position it achieved in the domestic market), and of Autodesk's failure to field a credible product in the entry-level two-dimensional drafting market, as suggested five years earlier in the "AutoCAD Lite" proposal (see page 240).

This acquisition was a contentious issue at the last Autodesk Board of Directors meeting I attended in February 1989. Malcolm Davies, who argued vehemently in favour of this acquisition responded to my question, "Why won't this fail, like all of Autodesk's previous acquisitions" with the assertion "Because I won't let it fail." Let history judge.[296]

March 7, 1989
For Immediate Release

AUTODESK TO ACQUIRE GENERIC SOFTWARE, INC.

SAUSALITO, Calif.—Autodesk, Inc. announced that it has signed a letter of intent to purchase privately-held Generic Software Inc. The acquisition is scheduled to be finalized within the next 30 days.

Autodesk's AutoCAD(R) product is the industry standard computer-aided design (CAD) software; it is in use at more sites worldwide than any other CAD product. Generic Software, a market leader in low-cost CAD software, and Autodesk together would have a combined installed base of almost 500,000 users.

"We plan to have Generic Software continue operations out of its Bothell, Washington, location," said Alvar Green, Autodesk's president. "The company has been very successful with its products and its current organizational structure and we see no reason to tamper with that success." According to Green, Bob Fulton, Generic Software president, will retain his management responsibilities at Generic.[297]

[296] When the acquisition of Generic Software by Autodesk was announced, a spoof version of the Generic Software newsletter front page was produced with the headline "The Empire Strikes Back" above a large, threatening "Darth Vader Mask" Autodesk logo. For months after the acquisition, Al Green kept a copy of this "front page" above his desk.

[297] Bob Fulton left shortly thereafter. In 1993, manufacturing and development activities were curtailed at Bothell, and the announcement was made that there would be no further development of Generic CADD beyond version 6.1.

Expressing his support for the acquisition, Fulton stated, “this is a winning combination which will lead to new opportunities for Generic’s products, particularly in the international marketplace where Autodesk has a well-established reseller network.” Autodesk’s Authorized Dealer network will now have the opportunity to carry Generic products, while Generic’s existing distribution channels will remain fundamentally unchanged.

- 30 -

Why Cellular Automata?

I met Rudy Rucker at the Hackers' Conference in 1987. I'd read most of his books, but I didn't know he'd gotten into computers. He brought a PC with a CAM-6 board, a hardware cellular automata simulator, and showed some amazing demos. After I got back from Hackers' I thought more and more about cellular automata, which I'd basically ignored since the game of Life craze in 1970. I wondered if whether a software-only CA simulator could be made sufficiently general and fast to eliminate the need for a $1,500 CAM-6 board.

In November 1988, Rudy Rucker came to work at Autodesk full-time and began working with me on such a product. In June of 1989, we shipped "Rudy Rucker's Cellular Automata Lab," the first title in the Autodesk Personal Science series. The product received excellent reviews and was profitable, despite receiving essentially no resources for promotion and being ignored by the sales organisation. It was terminated on November 30, 1993.

The following, from Chapter 1 of the CA Lab manual, are Rudy's and my explanations of why cellular automata are interesting and important to the future of computing.

Rudy's Answer

The remarkable thing about CAs is their ability to produce interesting and logically deep patterns on the basis of very simply stated preconditions. Just as the Mandelbrot set arises from the repeated iteration of the simple equation $Z = Z^2 + C$, iterating the steps of a CA computation can produce fabulously rich output. A good CA is like an acorn which grows an oaktree, or more accurately, a good CA is like the DNA inside the acorn, busily orchestrating the protein nanotechnology that builds the tree.

I feel that science's greatest task in the late twentieth century is to build living machines: *intelligent artificial life*, known as *a-life* for short. In Cambridge, Los Alamos, Silicon Valley and beyond, this is the computer scientist's Great Work as surely as the building of the Nôtre Dame cathedral on the Ile de France was the Great Work of the medieval artisan.

There are two approaches to the problem of creating *a-life*: the top/down approach, and the bottom/up approach.

The top/down approach is associated with AI (artificial intelligence), the bottom/up with CA (the study of cellular automata). Both approaches are needed for intelligent artificial life, and I predict that someday soon chaos theory, neural nets and fractal mathematics will provide a bridge between the two. What a day that will be when our machines begin to live and speak and breed—a day like May 10, 1869, when the final golden spike completed the U.S. transcontinental railroad! The study of CAs brings us ever closer to the forging of that last golden link in the great chain between bottom and beyond. If all goes well, many of us will see live

robot boppers on the Moon.

A heckler might say, "Sure that's fine, but why are CAs needed? Why have a bottom/up approach at all? What do mindless colored dots have to do with intelligent artificial life?"

For all humanity's spiritual pretensions, we need matter to live on. And CAs can act as the "matter" on which intelligent life can evolve. CAs provide a lively, chaotic substrate capable of supporting the most diverse emergent behaviors. Indeed, it is at least possible that human life itself is quite literally based on CAs.

How so? View a person as wetware: as a protein factory. The proteins flip about like John Holland's genetic programs or like A. K. Dewdney's flibs; generating hormones, storing memories. Looking deeper, observe that the proteins' nanotech churning is a pattern made up of flows and undulations in the potential surfaces of quantum chemistry. These surfaces "smell out" minimal energy configurations by using the fine fuzz of physical vacuum noise—far from being like smooth rubber sheets, they are like pocked ocean swells in a rainstorm. The quantum noise obeys local rules that are quite mathematical; and these rules are in fact very well simulated by CAs.

Why is it that CAs are so good at simulating physics? Because, just as in physics, cellular automaton computations are i) parallel, ii) local, and iii) homogeneous. In both physics and in CAs, i) the world is happening in many different places at once, ii) there is no action at a distance,[298] and iii) the laws of nature are the same everywhere.

Whether or not the physical world really is a cellular automaton, the point is that CAs are rich enough that a "biological" world could live on them. We human hackers live on language games on biology on chemistry on physics on mathematics on—something very like the iterated parallel computations of a CA. Life needs something to live on, intelligence needs something to think on, and it is this seething information matrix which CAs can provide. If AI is the surfer, CA is the sea.

That's why I think cellular automata are interesting: A-life! CAs will lead to intelligent artificial life!

Rudimentary CA a-life already exists in the form of Brain's haulers, Vote's oscillators, and such classic Life patterns as Gosper's glider gun.

In the 1970s, Berlenkamp, Conway, and Guy proved that putting a lot of these objects together can make a universal serial computer, such as a PC. Any serial computation can be done by a CA, and any CA computation can in turn be done by a serial computer—in support of this last point, note that all the programs on this disk are serial programs written in C, Pascal, BASIC, and/or 8086 assembly language.

Many computations can be done much more rapidly and efficiently by a succession of massively parallel CA steps. And one does best to use the CA intrinsically, rather than simply using it as a simulation of the old serial mode—emulating an Intel chip by using a galaxy-sized array of blocks and glider guns is not the way to go. No, when we use CAs best, we do not use them as limpware animations of circuit diagrams. While behaviors can be found in top/down expert-system style by harnessing particular patterns to particular purposes, I think by far the more fruitful course is to use the bottom/up freestyle surfing CA style summed up in the slogan:

Seek Ye The Gnarl!

New dimensional CA hacks are possible, new and marketable techniques of parallel programming are lying

[298] Richard Feynman seems to say that CAs cannot be a perfect model for the world because nature does in fact have action at a distance in the form of the kind of quantum mechanical synchronicities discussed by the Einstein-Podolsky-Rosen paradox. But it is possible to write programs similar to CAs in which the cells do occasionally look at distant neighbors.

around waiting to be found, both in the form of individual CA structures and in the form of wholly different rules.

CA structures are labile and breedable in three senses: one can collide and interface different local patterns within the framework of a fixed CA rule, one can combine globally different CA rules (or ideas about them) to produce wholly new ecologies, or one can "gene-splice" the logic of successful rules. Then, like Alexander von Humboldt in the Americas, one botanizes and zoologizes and mineralizes, looking for whatever artificially alive information structures can be found in the new worlds. As always both top/down and bottom/up approaches are viable. We use bottom/up to find new ecologies and their flora and fauna. We use top/down to seed a given instance of a particular ecology with the sort of gene-tailored computer agents we want to breed.

In my own bottom/up searches I begin simply by hoping that my programs will display interesting output for a long time. Then I begin to hope that my programs will be robust under varying initial conditions, and that they will be reactive in anthropomorphizable ways. Once the program is, at this very rudimentary level, artificially alive, I may cast about for applications in some practical domain.

I think the most productive near-term applications of CAs are to image generation and image processing. A cycle or two of Vote, for instance, can be used for easy image cleanup, munching down all stray "turd bits". This technique, known as "convolution" in the literature, is used every day by NASA's massively parallel computer in Beltsville, Maryland, to process terabyte arrays of satellite photo data. Present-day designers of the newest commercial paint and graphics packages for the VGA will be putting CA rules into their image processor toolboxes. (Look, for instance, at what Border does to Dr. Tim's face.)

In the area of original image generation, I predict that one of the next big commercial computer graphics fads will be CAs. How about a logo that instead of being chrome is matte and luminous, with a smooth curved surface made of tiny moving mosaics of light, lightbits that form the crawling dirty haulers of Brain or the psychedelic shudder of Rug? These are what the expressive "flickercladding" skins of the robots look like in my two a-life science fiction novels, *Software* and *Wetware*.

Many simulation applications exist as well. The idea is to find a CA rule that looks like something you want to model. If you are lucky there will be some common underlying mathematics between the two. The Rug rules, for instance, are difference method solutions of the same differential equation, the Laplacian heat equation:

$$\frac{\partial^2 Q}{\partial x^2} + \frac{\partial^2 Q}{\partial y^2} = 0$$

This means, e.g., that a fine-grained Rug rule inside a fixed circular boundary set may serve as a viable model of a vibrating drumhead!

A last current application of CAs is to encryption. Either a CA can serve as a cheap source of "essentially random" encryption bits, or the whole message can be fed to a reversible CA. Stephen Wolfram claims actually to have patented the one-dimensional rule with Wolfram code #30 as part of an encryption scheme.[299]

But to recapitulate, *the real reason for studying CAs is to promote artificial life*. The most important use for cellular automata will be as "universes" or "arenas" in which to evolve better fractals, flibs, core-warriors, neural nets and expert agents, using gene-splicing, mutation, and our own "divine interventions" to achieve

[299] U.S. Patent Number 4,691,291, "Random Sequence Generators", granted September 1, 1987. Wolfram has also patented a hexagonal-cell cellular automaton for simulating systems described by partial differential equations; see U.S. Patent Number 4,809,202, granted February 28, 1989.

a rapid and dramatic evolution in these parallel processes. CA workers need your help in accomplishing the manifest destiny of mankind: to pass the torch of life and intelligence on to the computer. There are no more than a few hundred active workers in the CA field today;[300] *twenty-first century technology will need thousands more!*

© 1989 Rudy Rucker — Reprinted by permission.

John's Answer

Physics is local. The two great pillars of Twentieth Century science, general relativity and quantum mechanics, can be viewed as supplanting the mysticism of "action at a distance" and "force fields," by such mundane, self-evident, and intuitive mechanisms as the Riemann curvature tensor, virtual gluons, and the Higgs field. Both of these theories describing the universe in the large and the very small (albeit in mathematically incompatible ways), tell us that all the complex fabric of events we observe are consequences of individual particles locally responding to conditions directly affecting them, whether moving along geodesics in curved spacetime or undergoing interactions through particle exchange. Both theories have withstood all experimental tests to date, including many thought impossible when they were originally propounded.

A cellular automaton (CA) is a mechanism for modeling systems with local interactions. A cellular automaton is a regular lattice of cells with local state, which interact with their neighbors subject to a uniform rule which governs all cells. The neighborhood (the set of cells whose state can affect a given cell at one instant) can be classified by the dimensionality of the automaton (most experimentation is done with one- or two-dimensional automata), and by the geometric fashion in which cells are interconnected.

The rule is the "program" that governs the behavior of the system. All cells apply the rule, over and over, and it is the recursive application of the rule that leads to the remarkable behavior exhibited by many cellular automata. When experimenting with cellular automata, one is primarily engaged in defining new rules which lead to interesting or useful behavior. The programs in the Rudy Rucker CA Lab from Autodesk are tools for the person engaged in such experiments. Our programs allow you to create a rich set of rules and experiment with their behavior without requiring the purchase of expensive special-purpose hardware.

Cellular automata appear to be abstract and devoid of practical applications, much as was said of computer graphics not long ago. If you want to model a universe which seems to be made up of particles which interact locally, there are two basic ways to go about it. The first is to create a huge array of numbers that represents the interacting items, then find the biggest number cruncher you can lay your hands on and set it gnawing away at the problem. The supercomputer boom, fueled by applications of this approach to weather prediction, computational fluid dynamics in the transonic and hypersonic regimes, plasma dynamics, and an almost endless list of other applications testifies to the effectiveness of this approach.

But maybe there's another way. Until recently, cellular automata were primarily a theoretical tool. The price of a cellular automaton with uniform edge size increases as the nth power of its size, where n is the dimensionality of the cellular automaton. This gets out of hand rapidly, even if you're only working with two dimensional cellular automata. Therefore, although they may be the way the universe is really assembled and therefore worthy of study, no one would consider building one!

Hardware realizations of cellular automata, such as the CAM-6 board, have been built. The CAM-6 is not a true cellular system; it emulates one by using a fast RAM array and a look-up table, but it permits exploration

[300] The interdisciplinary conference CA 84, held at MIT in early summer of 1984, attracted some 150 participants.

of a rich set of cellular automata with performance adequate to study their behavior in detail. The CAM-6 is a highly effective tool, but it is, at $1,500, an expensive one. It's priced out of the reach of many creative people who should be exploring cellular automata. It was the desire to make cellular automata experimentation available at a low price to a large number of people that spurred the development of this product.

For cellular automata need only to find a concrete, compelling application to a real-world problem to burst into silicon and totally change the way we think about computing. Consider this: inside the computer you're using now are ranks and ranks of RAM chips. A 256K$\times$1 static RAM chip has a memory cell consisting of four to six transistors, connected in rows and columns to circuitry on the periphery of the chip. Even when the computer is running flat-out, you're using precisely one cell at a time. This is the classic bottleneck in the von Neumann computer architecture (John von Neumann was very aware of this problem; in fact, he and Stanislaw Ulam *invented* cellular automata precisely as a tool for modeling complex systems), which has led to proposals such as Backus' functional programming, neural systems, and many other architectural proposals, such as SIMD machines, which seem to be more effective in generating Ph.D.s than numbers.

If a two-dimensional cellular automaton with 256K cells were realized in silicon, it could compute 262,144 times faster than a serial processor accessing data bit-by-bit from a memory array. Yet, engineered for volume production, made in comparable volumes, and given time to slide down the learning curve, it need cost no more than a RAM chip. This is the potential of cellular automata. The beauty of two-dimensional cellular automata is that they map perfectly into our semiconductor manufacturing technology: they need the things it does best as opposed to, say, neural systems where the number of connections exceeds the capability of two layers of metal.

If there is merit in Edward Fredkin's suggestion that the fine-grain structure of the universe is really a cellular automaton, then cellular automata machines will play the role of particle accelerators in exploring this level of reality.

Some of the brightest minds of our century have been involved with cellular automata because they comprehended what cellular automata can do. John von Neumann, Stanislaw Ulam, John Horton Conway, Stephen Wolfram, and Edward Fredkin do not spend their time on nonsense. With the Rudy Rucker *CA Lab* from Autodesk, you can begin to explore the potential that attracted those men to create and research this new way to compute. And perhaps you will discover something that will add your name to the list.

There's plenty to discover. Rudy's semitotalistic 16-state RC program permits you to create $16^{16\times9}$ different rules for cellular automata. This is a number slightly larger than 10^{176}. My more general 256-state CA program lets you program $256^{2^{16}}$ distinct CA rules, which is a number larger than $10^{157,826}$. These numbers are "effectively infinite". Roughly 10^{17} seconds are thought to have elapsed since the big bang ushered in the universe. If you had been around since then, creating, testing, and evaluating one rule per second, you still wouldn't have made a dent in this number, anymore than a buck makes a dent in a trillion dollars. Take away one dollar and you still have about a trillion. Even with enough time, you'd have a lot of trouble writing down the results of your exhaustive search for rules, as the universe is believed only to have on the order of 10^{80} particles in it, so you'd run out of places to make notes even if you turned the entire universe into a cosmic all-encompassing Post-it note. If the still-unconfirmed Grand Unified Theories are correct, by the time 10^{40} seconds have passed, more than half of the protons will have evaporated into little poofs of energy and leptons, taking with them the fruits of your labors, and leaving what's left of you with $10^{157,826}$ bottles of beer on the wall.

So get started, people! The human mind works a lot better than blind search (one imagines a middle manager reporting, "Nothing much yet—we need more monkeys, more typewriters."). *CA Lab* unleashes your creativity

in a virtually infinite domain, where your discoveries may be not only interesting or rewarding, but may create a whole new world. The challenges in cellular automata are clear: how to realize them in hardware, how to apply them to useful tasks, and how to make money doing it. You now possess a tool for exploring all three.

Understanding AMIX

The American Information Exchange (AMIX) is perhaps the most innovative venture in which Autodesk has become involved to date. Without doubt is has been the hardest to explain. Autodesk invested in AMIX in June of 1988, with the intent of implementing and launching the world's first on-line market for information and expertise. Within a year, while the development of the server and access software was still underway, it seemed like almost everybody had forgotten the opportunity that existed and why we had become involved. I wrote this memo as a reminder and to persuade Autodesk to maintain its commitment to AMIX.

To: Al Green, Malcolm Davies, Ron McElhaney, Greg Lutz
From: John Walker
Date: September 7th, 1989
Copies: Eric Lyons, Phil Salin, Kern Sibbald

Subject: Understanding AMIX

I've been peripherally involved in several recent discussions about the development status of AMIX and the soundness of Autodesk's ongoing commitment of resources to AMIX as approved at the end of fiscal year board meeting. Some of these discussions have indicated to me a lack of complete understanding of the fundamentals of the AMIX business, and consequently of the nature of the risks and potential rewards of our investment therein. This note is an attempt to dispel, as best I can, any confusion about AMIX by stating, as clearly as I can, the essentials of the business opportunity it represents.

First, let me make explicit my biases. At the time we decided to fund AMIX I believed it was a sound investment opportunity, more for the public service than for the in-house product. I continue to believe this. For AMIX to succeed, it must run the gauntlet of any new venture; it can fail technologically, in its marketing and sales plans, or just because the time is not right. The considerations that apply here are identical to any efforts we undertake other than self-evident derivatives of current products which are inherently incapable of generating major increments in revenue. AMIX lacks the technological excitement of being a "neat hack" in the sense of Autodesk Animator or The Eagle Project. Instead, I believe it is a Good Business, in the sense of Jiffy Lube or Federal Express; one that may be properly positioned in time and the development of technology to grow and prosper beyond the estimates of its creators.

The Information Age

To call our time "The Information Age" is so obvious it has become a cliché. Our Information Age is the age in which mouthing of the hoary maxim "knowledge is power" is supplanted by the reality that "knowledge is wealth". The success of Autodesk and other software companies which have built businesses upon intellectual property—the product of the human mind, with minimal financial investment, physical plant and other capital equipment, are testimony of the advent of the information age. My paper "The New Technological Corporation" explained the ways in which I believe that intellectual property and its consequence, technological leverage, have scouted a new course to the summit of business success.

Signs of the emergence of the information age abound for the acute observer. Warren Buffett, the shrewdest investor of our age and foremost believer in "no-tech" investment; the man who proudly proclaims that he still doesn't know what a transistor is, buys a major stake in ABC/Capital Cities because he believes that control of channels of information is capital just as much as ownership of a steel mill. The acrimonious battles over patent and copyright protection of algorithms, software, "look-and-feel", and chip architectures are in fact nothing more or less than the gropings of a system designed for machines of steel and steam forced to come to terms with silicon and software.

As intellectual property supplants physical property, the ability to obtain access to accurate, timely information has become one of the keys to success. Our world is increasingly webbed by many different means of delivering information: some old, some new. In a given week I usually partake of the following:

Newspapers	(Wall Street Journal, New York Times)
Magazines	(Aviation Leak, Electronics)
Scientific Journals	(Nature, Science, IEEE Computer, ...)
Newsletters	(HSL, Dow Theory Letters, Foresight, Commercial Space, ...)
Television	(McNeil Lehrer, Nightline)
Radio	(BBC World Service)
Computer Broadcast	(USENET NetNews: comp.graphics, comp...)
Computer Online	(CompuServe CIS, Dow Jones News/Retrieval)
Computer Mail	(Technet, USENET, Internet)

All of these give me *information*. Information keeps me from being a dull boy, points out things worthy of further study and the like, but seldom addresses my chief need when I have a goal at hand and a pressing deadline to achieve it. I do not need *information*, I need *answers*.

Markets

We live in what is supposed to be the exemplar of a "free market economy" yet virtually nobody, even many who have achieved great success in business, understand the basics of the dynamics of a market. Markets cannot be easily analysed. They are simply places where the collective wisdom of the buyers and sellers of the moment duke it out and somehow, in a chaotic yet historically effective process, arrive at prices and hence values for the goods they exchange.

Markets are so messy and confusing that it takes an intellectual leap to propose a market as the solution for anything. Typically a dealer and/or retail distribution system will predate a market. For example, stock options

were available in a dealer market for decades before the emergence of exchange-traded stock options in the 1970s. Although futures markets for grain, cotton, and sugar existed in the 19th century, livestock futures were considered a radical innovation in the 1950s and 1960s, as were financial futures in the 1970s and index futures in the 1980s. Each of these developments created an efficient market in which prices could be assigned to commodities traded in far less visible ways before.

Markets have evolved, then, from concrete commodities such as a carload of wheat, to abstractions such as a 100 shares of Consolidated Diversified Industries, to concepts such as December XYZ 40 Calls, to financial vapourware such as options on futures on international shipping rates.

And yet, we have no markets for information.

Information Markets

AMIX is nothing more and nothing less than an attempt to create the first efficient market for information. If its current prospectus and goals seem modest, I suggest this is not a consequence of the potential of the venture but rather of the belief on the part of its principals that the concept is so robust that it can establish itself as a viable, self-sustaining business even with a very limited initial scope.

The information age has endowed information—knowledge, with the value previously reserved for physical property. Yet, as usual, our institutions—the software of civilisation—have lagged in evolving an efficient market in which this new commodity can be exchanged and hence assigned a value. AMIX is such a market. It was conceived and designed by people with a deep understanding of the role of markets in an economy and of their functioning in valuing abstract commodities.

To the person seeking an answer, existing online services provide plenty of information but few answers. To a potential provider of answers, there is no market to which he can bring his knowledge to offer to those in need of his talents. In economic terms, the market for information and answers is inefficient; it has high transaction costs for both buyer and seller and fails to provide the valuation feedback to both which is essential to make informed decisions. AMIX is an efficient computer-mediated market for matching buyers and sellers of information. The utility AMIX provides to the Information Age is as self-evident as the function of currency futures in the post-Bretton Woods era, yet equally difficult to invent and understand before the concept becomes so commonplace it's considered trivial and obvious.

Obvious Markets

In May of 1968, I visited the Battelle Institute in Columbus, Ohio and first heard a description of the "Xerox Necessity". What was the size of the market for perfect copies of black and white documents, produced on plain paper in less than 60 seconds, the year before anybody realised such a thing was possible? Effectively zero, of course. What was the size of that market 5 years after the appearance of the Xerox 914? Look it up.

What was the size of the market for overnight small package and document delivery in all the years before Federal Express figured out how to make such a service work, and gained the confidence of their customer base that the letter would actually be there the next day? One can imagine the entrepreneur who conceived of Federal Express fielding questions such as, "How is this different from First Class Mail?", and "Even if you succeed, why won't the Post Office jump in and put you out of business?". And yet. . . .

The rationale for our investment in AMIX is the belief that AMIX is the first well-conceived efficient market of the information age. Those who have created markets for the predominant commodities of the past have done very well, indeed. If this is, in fact, the information age, then eventually there will emerge an efficient market for information; one that supplants the retail or dealer market of existing information utilities with the give and take of an exchange floor and thereby assigns correct values to information, reduces the price of information to the purchaser, and opens the market to many more sellers, to the benefit of all and to the profit of the market maker.

Non-critical Issues

It seems to me that our concerns and evaluations of AMIX have often focused on issues that appear peripheral. The implementation of AMIX has encountered its share of the unforeseen technical delays that befall any engineering project. The delays which have ensued, and the way in which the problems have been addressed, are on the order of our experience in such projects as AutoCAD 2.6, Abbey Road, and AutoCAD/Macintosh. Let's not confuse the inevitable technical difficulties in realising a complex project with our belief or lack of belief in the value of the goal we're working toward. In all of the AutoCAD projects I named, at no point would we have considered a schedule slippage as grounds for cancelling the project as long as we believed in the product we were developing. So it should be with AMIX. If we understand and believe in the goal, then manageable schedule slippages should be accommodated. If we believe the goal to be unachievable or illusory, then we should expend no further effort or money in its pursuit, regardless of the state of its realisation.

I do not intend by any of this to demean the demonstrated competence and manifest dedication of Chip Morningstar, Randy Farmer, and the other members of the AMIX development team. They have undertaken a difficult task and are well on the way to completing it. I think, though, they would be the first to agree that while failure to complete the software development tasks they're doing can spell doom for AMIX, success can only grant AMIX a chance to succeed. I am confident that the AMIX development team will complete a product within an acceptable time and budget, albeit having encountered slippages on the order of those experienced by our other development projects, adequate to grant the AMIX business concept a fair trial in the marketplace. Consequently I'm not particularly concerned with the details of AMIX software development.

Why AMIX?

To invest in AMIX is to believe that we truly live in the information age; to accept that in our age information is a commodity as tangible as wheat, live hogs, Swiss francs, or the S&P500 index; that a self-regulated market in which information can be exchanged will inevitably emerge and prosper just as markets have developed throughout history for whatever was valued in that era; and that embodied in AMIX is the basis for the first true efficient market of the information age.

I believe in all of these things. I think AMIX was a sound investment when we funded it, and I believe it merits continuing our investment as planned.

I hope we will see AMIX through to the ultimate verdict of the marketplace, the information market that AMIX hopes to create.[301]

[301] AMIX began pilot public operation on January 2, 1992, and planned, after a market-building and software shakedown phase, an official launch of the service in the fall of 1992. On August 21, 1992, Autodesk announced it was divesting AMIX (see page 810). Deprived of the funds needed to roll-out the service, AMIX remains, at this writing, an idea yet untested.

Batting .300

It's one thing to talk about "try a lot of things and get behind the ones that work," but it's a lot harder when the time comes to make the cut and pull the plug on the ones that aren't working, especially when you realise that you're not talking about abstract numbers, but peoples' jobs, careers, and personal belief in something that didn't pan out. The decision to abandon AutoSolid as a stand-alone project and, instead, integrate it into AutoCAD was particularly difficult because it meant we'd be closing the Atlanta office and transferring the key development team to Sausalito. Other folks in Atlanta simply lost their jobs.

As if this wasn't tough enough on everybody concerned, right when Autodesk was incurring the expenses in moving people and closing the office in Atlanta, we had a particularly tough quarter and had to squeeze everything to meet our earnings per share expectation, including cutting back the quarterly profit-sharing bonus, which many employees had come to see as an integral part of their salary.

Lest disappointment turn into discouragement or, worse, enmity directed at those who had been doing their best for years to make AutoSolid a success, I decided the time had come once again to remind everybody that in business, as in baseball, what matters is the score at the end, not whether you hit a home run every time.

Statement for the Autodesk Monthly Meeting

by John Walker

September 8th, 1989

You know, it takes leather balls to get up here are try to out-do Richard Cuneo[302] in a sports analogy. So I brought one.[303]

Every now and then you run into something that's a little difficult to say and often it's easier to just keep your mouth shut. But I'm no Dodger. So, I'll just jump into the septic tank and start bailing.

[302] Richard Cuneo, a life-long Dodgers' fan—the *real* Dodgers—not those guys in L.A.—often used baseball stories to spice up his talks at company meetings. One time he brought a bat to the meeting and talked about "teamwork" shortly after the movie *The Untouchables* appeared. He hadn't seen the flick, so he couldn't figure out why people were either giggling to one another or skulking towards the door.

[303] A baseball. At this time, the San Francisco Giants and the Oakland A's were leading the pennant race so the Bay Area was baseball-crazed. The teams would eventually meet in October in what was dubbed "The Battle of the Bay" but will be remembered as "The Earthquake Series."

People who haven't worked for as many really awful places as I have sometimes lose sight or take for granted some of the ways in which Autodesk is different, and I'm not talking about the bay views, the great spirit, or the running dogs of capitalism in the hallways. Consider that ever since this company went public in 1985, every quarter has seen us reporting rising sales and earnings, meeting or matching the expectations of the Wall Street analysts.

Most companies don't do this. They usually screw up well before 4 years have passed, especially if they're on a growth path as steep as ours has been.

If you don't realise just how difficult and rare this is, you're inclined to think something like, "Another no-hitter. I wonder how they'll do tomorrow."

The last quarter was a squeaker—one of the toughest ones we've ever faced. We and the competition are going at each other knowing that the stakes in the contest are survival. We're moving into new markets and we're having to develop new ways to sell our products. And I think the economy is beginning to slow down.

Nonetheless, we pulled it off.

But I don't want to talk about that. I want to talk about AutoSolid. I've heard some very depressing and poisonous comments recently to the effect that "we only got a small profit sharing check this month because of the Atlanta situation." We got a small profit sharing check this time because the company had a Hell of a time making our numbers and just squeaked it out.

What about AutoSolid? One thing I find I have to keep saying over and over again is that the nature of business is trying lots of things, finding out what's working and what's not, and getting behind and prospering from the ones that work. If you're too timid to take your lumps, you'll fail by never having found your next great success. If you're too arrogant to admit you're wrong, you'll fail by pouring your efforts into things that aren't working rather than shifting them to things that may.

If you walk up to the plate and fail to get on base 70% of the time, they call you a hero. Why is it so hard to understand that batting 300 and making your best effort every time is the secret to success in business?

We tried one thing with AutoSolid, and it didn't work. Now we're going to try something else, preserving most of work we've poured into it so far, and I think we'll end up taking over the solid modeling business, in time. But while there's pain associated with trying something that didn't work, feeling shame and assigning blame are the wrong things to do, for most things that don't work aren't because somebody did something wrong—it's because you didn't have all the information, the timing was wrong, or just plain rotten luck. Certainly, our colleagues in Atlanta demonstrated competence, dedication, and hard work that is a model to us all.

When you strike out, you walk back to the dugout and resolve to succeed the next time.

Over the next several centuries of Autodesk's history, I think we'll be fortunate to bat .300. If we do, our company will continue to be one of the great success stories in the world of business.

• • •

Second, I'd like to make a brief comment about the stock market. After the crash in 1987 when I mentioned at the company meeting that I'd been feeling very nervous about the market, several people who got blindsided in the crash came up to me and said, "well, I'd really appreciate if you told me the next time you're nervous."

I'm nervous.[304]

I don't give investment advice, and I don't want to discuss the stock market or argue various opinions about it—basically, it bores me. Everybody has to make their own decisions. All I'm saying is that if you're up to your neck in the stock market, make sure it's because that's where you want to be, rather than because you haven't thought about it recently.

[304]My reputation as a stock market prophet is far from perfect, but this was one of my better calls. The day I said this the Dow Industrials closed at 2709.54. A little more than a month later, on October 13th, "The Crash of '89" took the market down to 2569.26. A recovery ensued which recouped the losses for a while, but it wasn't until February of 1991—sixteen months later—that the market decisively rose above 2700 to stay—at least so far.

We'll Return, After This Message

This is a science fiction story I penned in December of 1989[305] *which captures, better than any narrative based in fact, the "everything is possible," "imagination-constrained" spirit of Autodesk in the late 1980's. Sales and earnings were blowing the top off the bar graph. Autodesk had launched ventures to position itself at the centre of a host of emerging technologies posed to benefit from the predictable evolution of computer hardware: multimedia, virtual reality, hypertext / wide area networks, information markets, solid modeling, and automated manufacturing. What was not obvious to me at the time was that Autodesk's commitment was skin-deep or less—in fact, fewer resources were allocated to launch these products, when completed, than had been expended by the cash-strapped nascent Autodesk of 1983–84. But enough angst; let's wallow together in the Fat Times, when the Hackers' Conference was described as the "Autodesk off-site meeting."*

When the foundations of everything you think you know shift beneath you, you can *feel* it. The night Art Crane and I found the Message, it felt like that moment at the onset of an earthquake when you realize the floor is really moving. Even after twenty years, I can't recall that night without seeing reality shimmer slightly, like the distant mountains on a hot day.

What we found that January night expunged a century's accumulated smugness about our place in the universe. And the funny thing is, we weren't even looking in the right place.

In December of 1997 it seemed as if humanity was well on its way to figuring out the universe. The discovery of the Proteus particle by de Vany, Trang, and Zweig handed the astronomers an all-in-one answer to the riddles of the missing mass and the solar neutrino deficit. The human genome project was winding down having yielded, if little understanding, plenty of data. The Soviet outer planet robots were flying in formation toward their Jupiter gravity kick, thence to Voyager's ports of call and onward to Pluto and Charon. The Shuttle was expected to return to flight any month. That glorious daylight supernova of 1996 was just beginning to fade from the nighttime sky.

I figured Crane might call. When I answered the phone on that rainy day after Christmas, I wasn't surprised to hear his usual request, "Cliff—I'm down here at the office. Could you come in and help me with something?"

Art Crane's a programmer, and a damned good one. But mostly, he is the Bach of the wild-ass conjecture. You can't spend ten minutes with him without hearing him suggest a new mechanism for speciation in biology, opining that the roots of monetary inflation lie in the domestication of animals, or wondering how we'd know if

[305] It's far from the first science fiction story I've written, but it's the first one I've ever *sold*, breaking a strike-out streak spanning decades. "We'll Return..."appeared in *MicroTimes* Issue #112, August 23, 1993.

just a *few* electrons had different masses. He writes every idea down in a little notebook he always carries, then he types them all in to his machine every night. I haven't asked him how big the file is. He reads everything, seems to know all, has opinions on any topic you can name but will gladly argue either side. He brooks no inaccurate facts or sloppy reasoning. He's the kind of person who'd be intimidating and unapproachable, if only he ever *finished* anything.

He isn't a flake. At least not all the time. When he gets a Big Idea, he's like a terrier. He grabs it and shakes it till it falls apart. Then he sniffs at the pieces. For some reason, whenever he has a Big Idea, I always seem to get involved. Not that I mind. Except for the stock market idea. That one I minded.

Christmas day, Crane and I had been invited to Hack Watkins' for turkey dinner. Later, as we watched Watkins' kids reduce their holiday bounty to pieces siftable through chicken wire, Art was holding forth on his latest idea—second-hand SETI. He was fascinated by pre-discovery observations. Galileo spotted Neptune and even charted it next to Jupiter in one of his notebooks. If the next night hadn't been cloudy, he'd probably have discovered it more than 200 years before Adams and Le Verrier.[306]

The year before, more than a hundred amateur astronomers took pictures showing the Orion supernova brightening before it burst into naked-eye visibility, but not one noticed it till after the fact. Crane said this was inherent in modern science; building big new machines and using them for bold searches was sexy and easy to fund, but rarely did anybody rummage through the dusty archives until something interesting had turned up in new data. He figured that if we ever received a signal from another civilization, we'd probably find dozens of others buried in the archives, easily located once we knew what to look for. "The facts, dear Clifford, are not in our stars, but on our shelves." he said, "Why don't we look there?" When I headed home around midnight, he and Hack were kicking ideas back and forth about how image processing tools might be used to identify intelligent signals.

I showed up at the office around sunset. Not a soul was there, and only a few programmers. Crane, who never rose before the crack of noon, rarely undertook serious work before six. I was surprised to discover him already in his office, surrounded by a midden of books, pieces of paper, and partially consumed processed food-like substances suggesting, by its height, that he'd been there for several hours. He turned from the screen as I walked in, "Cliff, glad to see 'ya. Look, we gotta get more crunch power on this job."

This was the week for it. The company always closed between Christmas and New Year's. It was a tradition they called the "Annual Week Of Rest," which was a fine joke because there was another tradition that the programmers would use a week devoid of managers, marketeers, meetings, and memoranda to try to out-do one another in huge bursts of concentrated effort to impress each other and just incidentally enrich their employer when they staggered back, bleary eyed, to another year of "regular work." Not that it was expected, of course—but all the computers were left running that week anyway. The Exalted Founder had started it 15 years ago and still kept at it, at least in theory. Nobody could tell, actually; he'd gone through some kind of comprehensional singularity and nobody understood anything he'd done in the last five years.

Back in '97 Xanadu still wasn't finished, but if you knew where to look and how to access it on the Net, you could get most of the machine-readable raw science data since 1970. The Net was installing a new software release over the holidays and had declared connect time and data transfers free between Christmas and January 5th to encourage users to test it. Art proposed to make the most of this. He planned to search all kinds of astronomical data, from the earliest radiotelescope sweeps to the downlink from the gamma ray imaging telescope in that converted shuttle tank with an algorithm he called the "annoyance filter."

[306] Galileo's observation of Neptune was discovered in a study of his notebooks in the early 1980's by astronomer Charles Kowal and Galileo historian Stillman Drake—Dan Drake's father.

"Whatever the message is, and however they encode it, it's going to be obvious, at least in retrospect. Besides," he said, "we have all the inarticulate bozos we need on Earth. There's no need to scour the galaxy for more."

He figured the message would be a picture, and that it would be deliberately made easy to distinguish from a noisy background. This was right down Crane's alley. A couple of years before he'd been obsessed with the idea of developing a program to discriminate television programs from commercials. He wanted to start a company to make boxes that paused VCRs when commercials came on. He considered it an artificial intelligence challenge, "If any idiot can fast forward past a commercial, why can't we design a program to zap 'em?" He spent the better part of a year's spare time tweaking and tuning his algorithm before abandoning it; toward the end it worked pretty well, but not good enough to sell—it recorded moody commercials and edited out climactic scenes of cop shows. It *did* zap all the car dealer ads.

He likened the problem to protective coloration. "If television is a medium that delivers entertainment at the price of advertising, then advertising and entertainment will co-evolve to become indistinguishable in time." But in SETI, he believed, the incentives were different.

"Signal to noise! Look, are you going to go before the Congress of Galactic Elders Subcommittee on Unessential Projects and try to justify spewing terawatts of soft-sell to the stars? Whatever they're sending, it's going to be obtrusive, blatant, shrill, noisy, coarse, and puffing. It will be calibrated to attract, to rouse, and to entice. Count on it... it's *advertising*."

He started by taking the commercial zapper and optimizing it for single still frames by training it on magazine advertisements. Then he wrote a front end that scanned bit streams and applied a Fourier transform to recognize scan-line encoded images. These he planned to link into a tool that could process several hundred megabytes of raw data per hour per machine, generating very few false positives for intelligent messages. When I showed up, he said it was "coming along."

There are several distinct phases in an Art Crane project. The first is "inspiration;" he'll be consumed, usually without warning, by a Big Idea. He'll corner everybody in sight, talking a hundred words a minute, filling white boards with diagrams, shoving a sheaf of yellow paper in people's faces, and otherwise explaining why what he just thought of is not only the most important project on the planet at that particular moment, but painfully obvious to any vertebrate. Art has a talent for seeing how tools can fit together to do things they weren't designed for. His ability to estimate the difficulty of all the tasks involved in his plans is more modest; Hack Watkins once said "The only constraint on Crane's armwaving about holes in his designs is that his fingertips can't exceed the speed of light."

"Coming along" means he's making steady progress, but doesn't have much to show for it. Eventually, he'll get an initial, crude version to work, achieving a characteristically grandiose milestone he calls "initial operating capability." Thereupon he invariably "goes ballistic" as he begins to realize all the things he can add to the initial version. "Ballistic" is evocative not only of Crane's disregard of external guidance, but also of the frenzy and round-the-clock concentration that characterizes his efforts, rendering him in this mode more a force of nature than a colleague. This ends at a point of exhaustion when he's run out of things to add to the product or can't sustain the kind of effort he's been putting in, whereupon he enters the "gory details" period as he attempts to clean up all the loose ends and render the result usable to people other than himself. Finally, he deliberately "throttles back," catches up on his reading, and comes within a standard deviation or so of a normal human being as he awaits the next Big Idea.

The progression of phases is as predictable as the Moon's, but their timing has none of the regularity of the cosmos, much to the exasperation of all who work with him. Still, when he is good, he is very good indeed.

Three times in the last decade he had singlehandedly come up with the concept and initial prototype of products that now collectively accounted for half the company's sales—each one stemming from a Big Idea unrelated to his regular work. This track record made management more than willing to endure his eccentricities and propensity to indulge in what Watkins called "art for Art's sake."

He picked up the pizza box to the left of his keyboard and carefully stacked it atop a pile of books to clear a workspace. Grabbing a yellow pad and pen, he explained what he needed from me. "We can do this, but we've only got nine days till the Net starts charging again. Remember that cycle sucker animal you built for the inlet problem? That's what we need to drive the filter."

For once, what he wanted wasn't that hard. Last year when we'd worked on a hypersonic fluid flow problem for the Japanese spaceplane, I'd partitioned the job to throw all the idle time on all of our workstations at it. This was easier. All I had to do was take the data stream from the Net and apportion it out to all the compute resources I could lay my mitts on with an eenie-meeine-miney-moe distributor. With 500 workstations rated at a gigaflop or better sitting idle until the new year, we'd be able to cull a lot of data in a few days.

I went down the hall to my office, turned on the monitor, and set to work. The linear flow of real time changed into programming hours, measured more by the accumulation of pop cans, pizza and chinese take-out boxes, and crumpled sheets of yellow paper than numbers on the wall clock.

Art continued to tune the annoyance filter while I developed tools to obtain raw data from the Net, translate them into a uniform format his program could read, and partition the compute job among the machines in the building. Hours passed, then days. I went home and slept when I was too tired to go on, and I presume he did too.

We'd both finished by Monday the 29th. When I arrived, Art had queued me mail saying "Ready when you are." I walked down the hall thinking how odd it was that I'd been collaborating on a project with him, working in the same building, but hadn't seen him for four days. When I reached his office, I realized this was a good thing; the remains bespoke a major ballistic episode. Crane's office resembled a centerfold from *Toxic Waste Monthly*. I suppose an archæologist could date the slices of pizza by the quantity of mold. I focused on the task at hand.

It took a couple of hours to integrate Crane's filter with my dispatcher. When we were done, we ran some tests to measure the total compute power we were getting, then some known data checks to make sure we hadn't messed anything up. Around midnight we were ready to start on live data. For the last several days I'd been keeping our Net link busy transferring files we wanted to search to our local server since we could process on-site data much faster than remote files across the Net. Also, I wanted to copy as much data as I could while Net access was free. I didn't have the time to be very discriminating in my acquisitions; if a Net site had some files of interest and didn't publish a huge amount of data, I just figured, "Grab it all. Let Crane sort it out."

I entered the command to start the root task. It cloned itself onto every active workstation in the building, then each copy started pulling files to be searched from the master task queue and commenced scanning them with Crane's annoyance filter. The search was singularly unexciting. My task monitor showed the volume of data we'd scanned so far, the percent done, and the instantaneous compute power we were using. There's a feeling of power that comes from knowing you're doing more calculations every minute than all of mankind did from *Pithecanthropus erectus* to 1950, but the novelty of even so remarkable a fact quickly pales into something akin to watching paint dry.

Every time Crane's annoyance filter triggered, my control program popped up a window with the image on both our workstations and filed the picture for later examination in a directory called WOW. The filter seemed

to be working quite well. We got an image about every hour—mostly scanned illustrations from journal articles erroneously filed in raw data directories. Art figured we could ditch them by looking for captions, but they were appearing so infrequently he decided not to delay the search to modify the program. Around three **a.m.** Tuesday we were both taken aback when we found our first genuine interstellar message: a spacecraft, two beings, and some coded gibberish. Art looked at the source information in the title bar and confirmed that we'd discovered the Pioneer plaque in a NASA Goddard planetary image archive. "Right message, wrong way," he muttered. We decided to call it a night. With the search on autopilot, we could dial in from our home machines and scan the images in the WOW directory whenever we wanted. I, for one, had no desire to spend any more time in Crane's office until the janitors came in next week and cleaned it with a fire hose.

Tuesday, I slept late, gave the plants their biennial sip of water, unloaded the dishwasher, and didn't get around to dialing in to see if anything had been found until close to midnight. I looked over the 20-odd pictures in WOW and found the usual selection of false alarms. I checked the login history and saw that Art was dialing in every couple of hours to scan the pictures. The company was closed until Monday the 5th, so we'd have the better part of a week to continue the search. I figured we could get through about a quarter of the obvious candidate data on the Net in that time. The audacity of two programmers employed by a medium-sized software company searching three decades' accumulated scientific knowledge for interstellar messages didn't occur to me. I doubt Crane was capable of entertaining such a thought.

With the search for intelligent life in the universe toiling away, needing only sporadic attention, I turned my attention to serious work. The company had bought one of the just-developed superconductive long range NMR scanners, and I was working on a prototype for a product I called "*Fantastic Voyager*" that let you fly around in a 3D image of your own body using the cyberspace gear we made. The job was challenging both from the standpoint of identifying the different tissue types from the NMR samples, and in developing navigation models to let you trace the paths of blood vessels and nerves through the body. Since my home machine had all the power and storage I needed, I worked there, as usual disappearing into the project and surfacing only sporadically to eat, sleep, or dial in to scan the WOW directory.

Wednesday night I stopped by Bart Lazslo's New Year's Eve party for a couple of hours and was amused to hear Crane grandiloquently describing our little holiday hack as the "Crane/Slatkin Quest For Galactic Intelligence." He related our discovery of the Pioneer plaque much more dramatically than I could have, or, for that matter, than the facts justified. He'd brought a portable computer and hooked it up to show pictures from the WOW directory on Bart's projection TV. He'd even programmed a scrolling subtitle, "LIVE from the accumulated wisdom of mankind." Much to the amusement of those irritated by Crane's antics, that night the fount of wisdom yielded up a scatter plot of globular cluster velocities, what appeared to be a calibration signal for an interferometry run, and, at the stroke of midnight, an order form for the *Proceedings of the 1993 IAU Vienna Workshop On Relativistic Jets*. Sue Hardiman went to the piano and composed a song on the spot. All I remember is,

> *Crane pursued the aliens,*
> *Through the archives on the Net.*
> *But the aliens came and kidnapped Crane,*
> *In their Relativistic Jets.*

When I left, they were still adding verses of monotonically decreasing quality. Crane was singing along lustily.

I spent the rest of the week in a happy blur, working on *Fantastic Voyager*. Every now and then Crane or I would notice something amusing in the WOW directory and send mail pointing it out, but otherwise I didn't

pay much attention to the search. By Sunday night, *Fantastic Voyager* was working pretty well. After tiring of flying lazy orbits around my pancreas, I finished up the documentation describing *Fantastic Voyager*. Another part of the Week of Rest tradition was that the day the company reopened, papers describing the programmers' projects would appear in everybody's mailbox and the software would be on display in the Demo Lab. After I queued the paper to the mail system, I drove down to the office around midnight to install *Fantastic Voyager* on the lab machine and make sure it worked there. Also, I wanted to double check that the Quest was set to shut down before 5 **a.m.** when the Net started charging again. At normal prime time rates, our data transfers for the Quest would've gone through my monthly salary every day, and I wasn't about to get docked a couple of weeks' pay due to a typo.

Crane was in his office, writing up his Week of Rest project, a redesign of an obscure internal algorithm that would speed up one of our products about 30%. He'd actually cleaned the place up, to the extent of even returning most of the books to the bookshelves that covered the walls. We chatted for a couple minutes, then I went down to the Demo Lab and he returned to his documentation. As usual, my demo didn't work the first time—I had to dial into my home machine and copy some files and rebuild. Anyway, it was almost three before I was satisfied it would work that afternoon.

I went back to my office and locked up, then stopped by Crane's on the way out. He'd finished and queued his document, and was idly flipping through the 150 or so pictures in WOW. He said we'd accomplished something worthwhile and suggested we should write a paper about the Quest. Every message we'd found, after all, was the product of intelligence. If we'd broadcast the Net feed to the stars, any civilization with our program would have found ample evidence of intelligence. He thought we might be able to persuade the investigators on the various radiotelescope projects to routinely run our filter over their data, forwarding all the images it found to us for examination.

We talked about how we should phrase such a paper and what journal might publish it, and so on, in the kind of lazy-days bull session exhausted programmers are prone to as they sit on a virtual veranda on the banks of the Mighty Megaflop and watch time and data flow by. "Peep." Another image popped up on the screen: a NASA logo. Crane hit the key to file it away. It was already four, so we decided to sit around and make sure the Quest terminated on schedule at 4:50. Nobody'd be expecting any programmers to show up much before two **p.m.**, so there wasn't any reason not to see it through. Our conversation was describing a chaotic orbit around the usual attractors of evolution, anarchy, programming languages, quantum electronics, and office politics, when we got another "Peep." It was 4:32:19.217 PST.

I can vouch for the time; it's in the execution log. My memory of what happened after the image popped up on Crane's screen is less precise. We glanced at the screen, then we stared at it. The image filled about a quarter of the screen; it was 2039×2053 pixels. Four bold black diagonal stripes set off each corner and continued as thin alternating black and white pinstripes along the edges of the image. Had the upper left stripe proclaimed "**New!**," the layout would've blended indistinguishably into *Time* or *Scientific Enquirer*. What was within the border was another matter entirely.

The figures on the right were humanoid, but not human. They were simply the best aliens I'd ever seen: there wasn't a single missing or disproportionate characteristic about them, but they were *all wrong* to be humans. There were two adults, male and female—the female was holding a little one. On the left was what was clearly a conversion table between binary numbers represented by "−" and "|" and eight digit symbols. The middle of the picture was divided into four boxes, three with a large central dot, curved lines, and nomenclature using the eight digits. The fourth contained what could only be an orrery presentation of a planetary system. At the bottom were four rows of 64 of the digits given at the left of the picture.

Crane wasn't a practical joker and neither was I, but after Lazslo's party all the usual suspects knew what we were up to, and half a dozen of them would enjoy nothing more than sending us on a merry chase. Art rolled his chair over to the screen to check the source. I looked over his shoulder. We looked at each other. Suddenly I knew how Penzias and Wilson must have felt when they concluded that they'd either discovered pigeonshit in their antenna or the birth cry of the Universe. What we were looking at was either the most brilliantly executed gotcha I had ever encountered, or, well... precisely what it appeared to be. Art copied the source identifier, "**HGIDB11-23@NIH.GOV**" into his notebook, tore out the page, and said "I think you'd better verify this across the Net, and don't use any software we've developed this week; it may have been tricked."

Just to be extra careful, I dialed into one of the commercial compute services the company subscribed to and used standard retrieval and image processing tools in that system's library to extract the image from the location Crane's filter had spotted it, accessing the read-only master published copy of the database across the Net. It took me two hours to be sure. When I returned to Crane's office, the eastern sky was brightening with dawn. It was going to be one of those windy January days when the air is so clear even the distant mountains seem close enough to touch. Yet on this morning, only the weather seemed clear. Crane had also been checking out the Message.

If I'd been more selective in choosing data to scan, we'd never have found it. There could be no doubt; the Message was real, and it was right smack in the middle of where it most definitely didn't belong. We had found it in the NIH Human Genome database.

Every human who'd turned his eyes skyward wondering, "Are we alone?" had been carrying the answer, all along, in every cell of his body. Since then the same Message has been found in every mammalian genome that's been dumped, yet nowhere else in the animal kingdom. That checks with the age of the Message. Three of the boxes in the middle of the Message are orthogonal views of the Local Group galaxies, centered on the nucleus of the Milky Way. Running their motion forward from the positions in the chart to the present day dates the Message at about 250 million years before the present: just about the age of mammals. The orrery indicates they came from the fifth planet of a star with six terrestrial planets and three gas giants. Which star? Nobody has a clue, and what with differential motion over an entire circuit of the Galaxy, any position in the Message would be worthless anyway.

Whoever wove that Message into our DNA meant it to last until we figured out how to read it—it's built inextricably into the protein expression mechanism so any organism with a corrupted copy won't be viable. It's obvious in retrospect. If a visitor wants to leave a message, why not make it a self-reproducing, error-correcting message that any sentient race would stumble upon as soon as they undertook to reverse engineer their own design?

Crane immediately suggested what's become the consensus hypothesis about the number at the bottom of the Message. He thinks it's the product of very large prime numbers and that when we manage to factor it, they'll be the key to decode the second Message. Two problems: first, we either need to learn a whole lot more about factoring, build computers a million times faster than the latest quantum electronic models, or else wait the tens of millions of years it looks like factoring that number will take.

Second, we have yet to find the other Message. Nothing more has been found in the genome, although Crane and many others suspect the factors of the key number may direct us to additional information there. The intensive radio SETI program sparked by the Message has proved barren. So have the planetary explorations of the past 15 years, spawned by the possibility that those who placed the Message may have left a calling card somewhere in the Solar System. More recently, the limitations of robots in seeking the unexpected has

triggered the rapid expansion of the human presence into space. If we've found nothing, we've learned much, and perhaps we've taken our first steps in the footprints of those who left the Message in our genes.

Crane believes that when we find the second Message and are able to read it, we'll have proved ourselves ready for the instructions it contains. He thinks it will tell us how to find those who left it, and how to go there. We'll return, after this Message, and I think they'll be proud of us. They were already proud when they put the Message there a Galactic year ago. Proud enough to have signed their work.[307]

[307] Years later (1992–1993), a *Star Trek—The Next Generation* episode featured a similar theme: the humanoid genome was the gift of a parent species, containing a message readable only when all their diverse offspring came together to assemble it. That plot owed nothing to this story and, causality remaining unimpugned by counterexamples, this story was conceived independently of that script.

Caustic Moment

For years, our goal in expanding the application programmabilty of AutoCAD was to reach the point where the facilities available to a third-party developer to extend the functionality of AutoCAD equaled those of an Autodesk programmer with access to the source code of AutoCAD. For my "Holiday Hack" of 1989–1990, I decided to build upon the success of the Eagle Project and actually remove a current feature from the source code of AutoCAD and transform it into an ADS application, creating, in the process, a library that mapped internal AutoCAD subroutines into their ADS equivalents and thereby encouraging the emigration of other components of AutoCAD.

This project was ultimately stillborn, but it paved the way for the future evolution of AutoCAD. Most major feature enhancements introduced by Autodesk since 1990 (Advanced Modeling Extension, Advanced Visualisation Extension, Advanced Data Extension, AutoSurf, etc.) have been implemented as ADS applications built atop the AutoCAD database and user interface. In this sense, The Eagle Project and Caustic Moment were the pathfinders for the evolution of AutoCAD for years to come.

Caustic Moment

Emulating the database and geometry facilities of AutoCAD allows removing the IGES translator to an ADS application without extensive modifications. This development may herald the end of the growth of the AutoCAD core.

by John Walker
January 1st, 1990

TEN YEARS AGO today, I was celebrating not just the dawn of the 1980's, which I hoped would be an improvement over the Souring Seventies, but also the first successful multi-user test, achieved minutes before midnight, of NOS/MT, the Unix-like operating system that seemed to be the future of Marinchip, my company at the time. What was to eventually become AutoCAD was, on that night, less than 2000 lines of undocumented code in an obscure language for an unpopular computer.

Eight years ago, I was beginning to sketch out the plans for starting a software company from the bones of Marinchip, wondering what products might carry it to success in the marketplace. Developing the CAD product, which was at that time Marinchip's flagship, was high on the list. By the summer of 1982, development of AutoCAD was underway in earnest. The very rudimentary facilities of the original system were being extended

in all directions, leading to the introduction of AutoCAD in November of 1982. At that time, the program had grown to about 12,000 lines of C.[308]

In the ensuing years, as AutoCAD has continued to evolve to meet the needs of its community of users, the AutoCAD core has grown, seemingly without bound. As of development release **`z.0.64`**, the core source plus include files totals 239,153 lines. AutoLisp adds 22,354 lines to the pile, and ADS contributes another 17,338. MS-DOS drivers account for 107,588 lines, and non-DOS drivers 18,048 lines. AutoCAD's special C library is 13,128 lines. So, the total source code contributing to a present-day release of AutoCAD is on the order of 417,609 lines—and that doesn't count tools, regression tests, the Macintosh tugboat code, the ADI kit, the OS/2 Presentation Manager interface and DDE library, and all the other stuff I undoubtedly forgot to count.

However you add it up, this is one big, hulkin' program. Further, all our efforts in removing drivers, then devolving them upon vendors through the ADI program, and providing user-customisability through menus, scripts, and AutoLisp have merely reduced the rate of growth of the AutoCAD core, which still accounts for roughly half the source lines in the program and continues to grow with every passing day. The core is a treacherous place, ruled by strange customs and conventions, including wild constraints on memory usage, files that aren't files (unless they be ASCII), C library routines that behave in unusual ways, and the ever-nagging question "what if this causes my overlay to be reloaded?". The lore of AutoCAD core code craftsman is unwritten, continually evolving, and very hard for an apprentice to master. All of this makes the process of extending AutoCAD to maintain and expand its market leadership increasingly difficult, painful, expensive, and time-consuming. What can be done about this?

> *Sharp the strife, violent the collision,*
> *Difficult 'twill be to reach a decision.*[309]

With the advent of ADS in 1989, it at last became practical to implement extensions to AutoCAD outside the core code without incurring unacceptable penalties in efficiency, user interface, and portability. The Eagle Project, launched in July of 1989 and scheduled to ship in mid-1990, demonstrated the feasibility of this approach and moved it into the mainstream of Autodesk's plans for AutoCAD. Additional projects such as the Applied Geometry NURBS interface and the Xanadu/AutoCAD front-end are also taking this approach. None of these efforts have been easy; each one has encountered shortcomings in the ADS interface that must be remedied by core code modifications before the project can succeed. Nonetheless, it is unarguable that each of these projects will be accomplished in less time, with less effort, suffering, and destabilisation of AutoCAD than had they been integrated into the AutoCAD core in the conventional manner.

> *The one will press with energy immense, the*
> *Other dodge with footwork fancy.*

Now that ADS has matured, tempered by the demands of these projects, it's time to consider the next logical step: not just freezing the AutoCAD core as-is and concentrating development resources outside it, but actually beginning to shrink the core by removing features from it which, had ADS existed when they were originally implemented, would scarcely have been considered as candidates for core code. This hopeful season is one of dismantling empires, of devolving structures that have proven inefficient and unworkable. What better time to put ADS to its most demanding test to date: the migration of IGES from AutoCAD into an ADS application.

[308]The distinction between core code and drivers came much later, when Greg Lutz implemented configurable drivers in version 1.3. AutoLisp was, of course, not shipped until January of 1986, although a precursor of AutoLisp called "Variables and Expressions" was shipped in May of 1985.

[309]This, and the rest of the Chorus to this document is taken from Aristophanes' *Frogs*, 405 **b.c.**, R. H. Webb translation.

A Brief History Of IGES

I never wanted IGES to be in the AutoCAD core. When I began the IGES project in July of 1985, I intended that IGES would be an extra-cost application, selling for say $500, rather than being included with every copy of AutoCAD. Not only did I hope by this to raise some extra revenue to fund IGES development and support, but, recognising from the start that IGES would be a difficult and messy chore, albeit an essential one, my intention was to limit the market for IGES to those serious customers who would use it for practical purposes and generate useful requests for change, rather than frivolous users writing bug reports every time an IGESOUT/IGESIN "lost something" a DXFOUT/DXFIN preserved.

Keep not always the same stance:
Attack however there's a chance.

In 1985, the only alternatives to implementing IGES within the core were as an AutoLisp application and as an external program that processed databases in DXF form. AutoLisp could be ruled out easily, as the restrictions on program size and execution speed were intolerable for a task as complicated as IGES translation. In addition, access to the entity database was not added to AutoLisp until Release 2.6 in June of 1986 (having been coded during the Week of Rest at the end of 1985), and access to AutoCAD's symbol tables, essential for IGES to obtain block definitions, line types, text fonts, etc., was not added until even later. A DXF-based translator would have certainly been feasible (indeed, several have been marketed by third parties), but it would have been far more cumbersome to use and difficult to implement. IGES translation, which already involves processing a huge ASCII file, would require an intermediate DXF file, also huge and slow to write (binary DXF not having been implemented until Release 10). In addition, many operations in an IGES translator require flexible and efficient random access to the AutoCAD database and symbol tables, and the support of a wide variety of geometry service functions. These routines are available within the AutoCAD core, but an external DXF-based application would be forced to create its own AutoCAD-like database from the DXF file and replicate all the processing facilities so readily at hand within AutoCAD.

Bring all your resources into play;
Wrangle, tangle, be flayed and flay.

Regarding this soberly, and remembering that IGES was initiated at the very end of the development cycle of what was eventually called Version 2.5, I concluded that the best way to get IGES done and out the door was by going ahead and implementing it within the core. After all, overlaying would prevent IGES from increasing the memory requirements to run AutoCAD, and we'd already decided to abandon support of floppy disc only machines as of Version 2.5, so growth of the on-disc size was not a major consideration. Integration of IGES in the core would not preclude marketing it as an extra-cost option; after all, at that time AutoCAD was offered in base, ADE-1, ADE-2, and ADE-3 trim levels, differing primarily in which overlay files were supplied on the release disc (actually, it wasn't that simple, but you probably don't want to know any more than that). With a little ingenuity and native cunning, we could figure out a way to sell an IGES package consisting of overlay files you copied into your AutoCAD directory to enable the IGESIN and IGESOUT commands. This would be packaged with the IGES document, and, if done in a sufficiently elegant (read sneaky) fashion, might even permit updates to IGES without the need to release a new AutoCAD.

Draw arguments old from out your store,
Venture subtleties never used before.

In the words of Dennis the Menace, "You can't tell how deep a puddle is from the top." Even though we walked into IGES knowing it was going to be difficult, IGES has never failed to surprise us with the amount of work needed to usably interchange drawings with other CAD systems, themselves only marginally compliant with the ill-defined and mutable IGES standard. With the camel's nose of IGES within the AutoCAD tent, IGES became a part of the mainstream of an AutoCAD release, at least in the sense that IGES bugs had to be closed out before an AutoCAD release could be shipped, even though the overwhelming majority of AutoCAD users never used IGES. However, IGES never became fully integrated in the minds of the developers, so that new features would be added to AutoCAD without implementing them in IGES. This would lead to crises in the final development phase of AutoCAD releases, when furious efforts had to be mounted to implement new IGES code, delaying the AutoCAD shipment date.

To compound the pain, what with everything else we had to do between mid-1985 and the shipment of Version 2.5 in 1986 (this was the first year after our initial public stock offering and was the period in which we were struggling to get the hardware lock ready for introduction with Release 2.5), we never managed to separate IGES from the AutoCAD product, foregoing the additional revenue this would have generated and making it impossible to distinguish an IGES user from any other AutoCAD customer. The following years have not seen IGES and AutoCAD coming to co-exist on any better terms. Virtually every major AutoCAD release has been delayed by an "IGES crisis" of one form or another, resolution of IGES bugs critical to obtaining business remains rigidly coupled to the AutoCAD release cycle, and the memory constraints imposed by the AutoCAD core have required the IGES code to become increasingly fragmented and convoluted and consequently harder to maintain.

Our decision to support IGES in 1985 was correct; IGES has become an essential part of a CAD system today, and with the growing focus on CALS, of which IGES is a part, will continue to be the price of admission to sales to government agencies, their contractors and subcontractors, and, increasingly, large industrial customers. We can only expect our commitment of resources to develop, maintain, and support IGES to increase as our business expands in those sectors. Any steps we can take to reduce the difficulty of IGES development will further the achievement of our goals. The whole smelly, dirty camel's in the tent, and his tail's flickin' at the 640K tentpole that holds the whole mess up. It's time to see if there's a better way to go about this.

Goals and Strategy

There's little doubt that IGES can be implemented as an ADS application. ADS in Let It Be, with the advent of **`ads_entmake()`**, provides essentially the same access to the AutoCAD database as exists within the core. The only major disadvantage an ADS program faces is the inability to create AutoCAD symbol table entries (for blocks, layers, line types, etc.) with a mechanism like **`ads_entmake()`**, but carefully chosen commands submitted with **`ads_command()`** can circumvent this shortcoming.

However, code written as an ADS application looks nothing like AutoCAD core code. The means used to access the database, to look up information from symbol tables, to inquire system variable settings, and to create entities are entirely different. Consequently, extracting a chunk of AutoCAD and turning it into an ADS application consists not just of the usual operation of severing connections and tying off bleeders, but virtually reimplementing all the portions of the code that access the database. In the case of IGES, that's about all there is. Although dwarfed alongside the AutoCAD core, IGES is a huge program in its own right; as of **`Z.0.64`** the IGES complex totals 14,103 lines of code. *This is larger than all of AutoCAD 1.0: core, drivers, the works!* Another purportedly arcane part of the core, the hidden line complex, is about a third the size of IGES.

Whatever the benefits of extracting IGES from the core, the prospect of what amounts to re-implementing 14,000 lines of code from scratch is not one that excites this programmer. After my earlier experiences with ADS suggested that ADS would support IGES, I began to investigate ways that IGES could be piecewise migrated from the present implementation within the core to a totally separate ADS version without disrupting the product development and release cycle, never requiring a total rewrite with attendant hiatus in IGES releases, one which would deliver the benefits of an ADS IGES as soon as possible with the minimum effort. This project is the fruit of that investigation; what I have at this point is very much a work in progress which will require substantial additional labours before the cutover from the current built-in IGES. However, the nature of the implementation allows this work to be done in parallel with the close-out of Let It Be, without disrupting that process in any way, and should permit shipment of an ADS extended IGES product well before the shipment of Release 12.

Implementation Notes

If you feel your audience uninitiate,
Unable profundities to penetrate,
Rest easy; out of fashion is naïveté.

The essential aspects of this project are as follows.

- Use the IGES modules from AutoCAD with as little modification as possible.
- Extract the self-contained core support routines needed by IGES.
- Develop a comprehensive emulator for the AutoCAD core database, symbol table, and system variable facilities.

Let's examine the rationale and some of the details of each of these.

Using IGES Modules

Since I envision Let It Be shipping with the current, within-the-core implementation of IGES, and the transition to ADS IGES occurring thereafter, and since substantial work remains to be done on IGES before Let It Be ships, I assigned a high priority to avoiding divergence between the built-in and ADS streams of IGES development until the built-in stream is frozen in anticipation of its discontinuation, presumably in Release 12. As long as divergence is avoided, the built-in IGES will continue to be the main development stream until Let It Be ships. Changes to this stream can be integrated into the ADS IGES stream simply by recompiling the updated files, resolving whatever new conflicts may have appeared by changes to the emulation libraries and, as a last resort, by **`#ifdef`** code submitted back into the AutoCAD development stream.

As an indication of the closeness between the core and ADS implementations of IGES, long **`DIFF`**s of all changes in IGES modules (both **`.h`** and **`.c`** files and including a couple of IGES bugs I stumbled over and fixed along the way), between the **`Z.0.64`** distribution and my current ADS IGES development directory total only 150 lines. Although some additional changes will be required to bring the ADS IGES up to full functionality, I do not anticipate the need for additional major code modifications in these modules.

Any conditional code required in the IGES modules is controlled by a new compile-time variable named **`OUTBOARD`**. This variable, never defined within the AutoCAD core, is intended to be used by code that exists both within the core and outside. I recommend that we define a variable, **`INBOARD`**, within AutoCAD, so that conditional compilation statements can be written in the most straightforward manner. My emulation library does not, of course, define **`INBOARD`**.

Extracting Core Routines

The AutoCAD core provides a rich set of tools for linear algebra, geometry, storage and list management, file I/O, and manipulation of objects used within AutoCAD. Any code removed from the core must either be provided with versions of these functions or converted to get along without them. Fortunately, most of the heavily-used service functions are reasonably self-contained and easily extracted from their homes in the core. As I worked to get IGES running outside the core, I simply made a list of the functions it used, and for all true support functions (i.e. those not used to access the database or other state-containing parts of AutoCAD), collected the required functions into a module I named **`autocore.c`** performing, as I went, transitive closure to pull in functions used by those I included.

At the moment, the functions implemented within **`autocore.c`** are:

`addvec alloc angle angle2 arbaxis aschand atwt cfclose cfunlink cndfre cross distan distan2 distan3 distsq distsq2 dot dotp eqv fabsv fcirceq fcirceq2 fixangle flineq fmax fmin fuzzeq g2pilc ga2p gd2p gdpl gdsq2p gl2p handasc hashandle hmabort identv intconic is1bcode is2bcode makerot matxmat mfv napln nulvec pconvt rswap scal setflags setpos sqabsv strsave subvec sumvec ucase unimat unisub univec vconvt vecxmat wtat`

Function prototypes, data type declarations, and other definitions required by these functions are in **`autocad.h`**, which is included in the compilation of every module extracted from the AutoCAD core. In addition to the routines collected into **`autocore.c`**, the following AutoCAD core modules are used, essentially unchanged: **`dxf.h`**, **`erstruc.h`**, **`erstruc.c`**, **`lstsubs.c`**.

The AutoCAD Database Emulator

> *Veteran campaigners of many a fray,*
> *The spectators come well-girt:*
> *Each has book in hand, each has wits alert.*

The functions in the file **`autosim.c`** emulate the facilities used by AutoCAD core code to access the database, symbol tables, and system variables. As with the functions in **`autocore.c`**, these were chosen simply because they were needed by the IGES complex; if this package is used to export other components from the AutoCAD core (such as DXF, for example), some additional functions will almost certainly have to be emulated. Currently emulated functions are listed below; functions marked with an asterisk are currently dummies, not yet emulated but unneeded by the portions of IGES exercised so far into the project.

apndr	Add object to entity database.
blcolor	Build block colour table.
curfit*	Fit curve to Polyline.
dapndr	Append entity to block definition.
delnudr	Delete object from database.
delnulots	Delete complex object from database.
dferec*	Define entity record.
dotblock*	Build dimensioning **DOT** block.
e3xform*	Transform entity.
e3xformi*	Build entity transformation.
e3xformr*	Reset entity transformation.
e3xformt*	Release entity transformation.
etrans*	Translate entity.
fnsplit*	Parse file name.
funlink	Delete file given descriptor.
getbd	Read block definition.
getlt	Read line type.
getly	Read layer definition.
getsm	Read arbitrary symbol table.
getts	Read text/shape item.
loadltyp*	Load line type definition.
makely	Set layer, creating if needed.
moddr	Modify entity database item.
nxtdr	Read next entity from database.
putbd	Write block definition.
putlt	Write line type.
putly	Write layer definition.
putsm	Write arbitrary symbol table.
putts	Write text/shape item.
randdr	Random read entity database.
rapndr*	Create dimension block.
rmoddr*	Modify dimension block.
scandr	Rewind entity database.
shpload*	Load shape definition.
shpnam*	Obtain name from shape index.
smsrc	Locate symbol table item from name.
snvalid	Test symbol name valid.
splinefit*	Fit spline to Polyline.
sposn	Return entity database position.
sqeptr	Seek to start of Polyline.
sseek	Set entity database position.
tdate	Obtain date and time.
tmpfopen	Open temporary file.
treadj*	Adjust text position.
txtbox*	Calculate text extents.
wrbptfopen	Open standard temporary file.
zoomax	Zoom to limits or extents.

In addition to these emulated functions, **autosim.c** contains the ADS initialisation and linkage logic used to invoke the IGES commands when loaded as an ADS application.

You're a sage and clever audience;
There's naught to fear, take heart, advance!

Loading A Database

Let's take a deeper look at how the AutoCAD core environment is emulated. When one of the IGES commands is entered, its command implementing function first calls **dbload()**, implemented in **autosim.c**. This function builds a simulated AutoCAD database in three steps.

System variables. First, **ads_getvar()** is used to capture the current settings of all AutoCAD system variables referred to by IGES. These variables are stored in static data defined in the file **autodata.h**, each named the same as its incarnation within the AutoCAD core. Other required information accessible through **ads_getvar()**, including the current drawing name and the AutoCAD version, are obtained as well.

Database index file. Since ADS provides all the facilities we need to read and write the AutoCAD database, there's no need to prepare a local copy of it. AutoCAD core code, however, addresses the database by means of **long** handles instead of the **ads_name** objects used by ADS. To avoid modifying the many places database addresses are assumed to be **long**s, **dbload()** makes a pass over the AutoCAD database and prepares a temporary file named **dbmap** which, addressed by an entity index, returns the ADS name for that entity. All entity references are translated through the file and the global symbols **entadr** and **entcnt** contain, respectively, an index into the file and its current length.

Symbol table files. Finally, the entire contents of each of AutoCAD's symbol tables are retrieved with the **ads_tblnext()** function. Each symbol table item is decoded into the format used within AutoCAD and stored in a temporary file named after the symbol table. Its file handle is kept in the local copy of the AutoCAD symbol table descriptor. The symbol tables are read and the copies prepared by **symtgrab()**, called from **dbload()**.[310]

As the block table is copied, the entities that define the block must be stepped through and their names added to the database index file. The index of the block definition entity is stored in the local copy of the block symbol table item. *This is not presently implemented, and hence IGESOUT of drawings containing blocks will fail.*

Using The Database

Once the local copy of the database is built, the various emulation routines access it straightforwardly. Since the local database is only used for the duration of one command, there's no need for it to be written in a portable format, so regular I/O calls can be used rather than the complicated mechanisms needed to maintain portability within AutoCAD. The **dbname()** function is called to map a **long** entity address into the ADS entity name used to retrieve or update it. Whenever an entity is read, it is obtained from AutoCAD with **ads_entget()**, then "scattered" into the fields of the **E** structure used within the core by the **scatlist()** function. The inverse process of collecting the values from the **E** fields for an entity and building a result buffer chain for **ads_entmake()** or **ads_entmod()** is performed by function **apmodr()**. When entities are added to the database, entries for them are added to the database index file and **entcnt** incremented.

Symbol tables are accessed by routines identical to those in AutoCAD's **smio.c**. Since local copies of symbol tables are kept, there is no need to call AutoCAD every time an entry is read. When a **putsm()** call is made to add an item to a symbol table, special code is required to submit the corresponding commands with **ads_command()** to build the table item. A local copy must then be added to the symbol table emulation file. *The current version of the IGES application does this only for layers; the final version will have to implement all symbol table objects created or updated by IGESIN.*

[310] There's a bug in AutoCAD **Z.0.64** that causes any attempt to retrieve the dimension style table to fail. This table cannot be copied by **symtgrab()** until the bug is fixed.

Closing The Database

When an IGES command is done, it calls **`dbclose()`**, which releases all the temporary files used to simulate the database. Since changes to the database are passed to AutoCAD as they occur, there's no need to pass any information in the temporary files back to AutoCAD at this point.

Changes to system variables made by IGESIN should be passed back to AutoCAD at this time. *This is not presently done.*

Development Environment

The ADS IGES application is developed in a self-contained directory, intended to be added to the AutoCAD standard build environment as a directory named **`iges`**, at a peer level to **`ads`**, **`lisp`**, and the others. There's a Unix **`Makefile`** in the directory which obtains its ADS libraries from the **`ads`** directory in the build tree. This code was built on the Sun, but there's no Earthly reason it shouldn't run on any machine that supports ADS, once the appropriate build procedures are defined.

At the moment, for convenience in developing and debugging, a composite application called **`iges`** is built that implements both IGESIN and IGESOUT. It probably makes more sense for the final customer version to supply separate **`igesin`** and **`igesout`** applications, as one should rarely need both loaded at the same time. The code in **`autosim.c`** and the **`Makefile`** supports this partitioning into separate applications, although at the moment doing so requires manually changing some **`#define`**s at the top of **`autosim.c`**.

Additional ADS Requirements

> *These clever men, how diligent!*
> *Here's another brand-new portent,*
> *Beyond the ordinary man's capacity;*
> *If told I'd doubt the teller's veracity*
> *And scorn the simpleton's naïveté.*

One benefit of developing ADS applications in-house is that the process helps to identify and remedy shortcomings of the ADS mechanisms before third party developers encounter them. IGES is very demanding, yet so far only one new ADS facility appears to be required. (I say "appears" because, as should be apparent from the many parts of this project cited in the text as still undone, more may lurk as consequences of emulation functions not yet written.)

The one unambiguous extension needed by IGES is an **`ads_textbox()`** function. This would parallel the **`txtbox()`** function within AutoCAD; given a text string, its height, style name, X stretch factor, obliquing, and mode flags, it would calculate and return (perhaps as two **`ads_point`**s with Z co-ordinates presently zero) the extents of the box enclosing that text string. IGES expresses text positions in that form, and I can't think of any way to obtain it other than reimplementing all of AutoCAD's text generation pipeline within the IGES application.

A "would be nice" feature that's not essential and may be provided for already in the ongoing development of ADS would be a signal to an ADS application at the first opportunity within each drawing editor session

(including that which loaded it, if not by **acad.ads**) when AutoCAD was quiescent and the application could submit commands. ADS programs such as **GRAVITY** and Eagle get around this with a little piece of code that's triggered when the first command they implement is entered, and that suffices for matters such as creating layers, blocks, and suchlike needed by the application. What I'd like to do with IGES is have the application, if loaded, **UNDEFINE** the standard **IGESIN** and **IGESOUT** commands, replacing them with those it implements (the built-in commands would remain accessible through their "true names"). I expect the ability for applications to transparently replace AutoCAD core commands will be a much-requested feature after ADS is shipped.

Also convenient, as additional AutoCAD commands are removed to ADS applications, would be a facility, perhaps implemented through the **ACAD.PGP** file, which could trigger the automatic **xload**ing of an ADS application when one of the commands it implements was entered. Any automatically-loaded applications would be automatically unloaded at the end of the drawing editor session that caused them to be loaded.

User Notes

Oh yes, how do you *use* this thing? First of all, very carefully. Getting the fundamentals of the simulation library working took much longer than I expected, so numerous loose ends remain to be cleaned up. Just because my demo works beautifully doesn't mean *you* can use it; a sufficiently advanced technology is indistinguishable from a rigged demo!

Load the IGES application with the command:

```
(xload "iges")
```

To create an IGES file of the current drawing, use the command:

XIGESOUT *filename*

To load an IGES file, bring up AutoCAD on a new drawing, then enter the command:

XIGESIN *filename*

Other than some diagnostic messages generated as the local copy of the database is prepared, the text screen output is identical to the built-in IGES commands (at least until it crashes).

Summary and Future Directions

Although the time available for this work permitted only a limited amount of the functionality of the built-in IGES commands to be debugged and demonstrated, the approach of transitioning IGES from the AutoCAD core into a standalone ADS application by initially emulating the AutoCAD core environment, permitting the existing well-debugged IGES code to serve as the base for development during the transition, has been tested and appears sound. This suggests a strategy for the future development of IGES something like what follows.

Let It Be continues development toward shipment without any change in direction. IGES changes required by that release are integrated into the core IGES modules. In parallel, the IGES subdirectory is added to the build environment and the support and simulation routines are placed there. The changes to the core IGES modules,

made conditional on **INBOARD** and **OUTBOARD** will be submitted as changes to the Z stream and carefully code reviewed to establish that they change nothing when built with just **INBOARD** defined.

It will then be possible to quickly and easily build an ADS version of IGES by copying the IGES files from the **coresrc** and **include** directories into the **iges** directory and invoking **make**. At this point it will be easy to continue development of the emulation libraries, aiming for identical operation of built-in and ADS-implemented IGES commands by the time Let It Be ships.

Accomplishing this will allow us to freeze the built-in IGES translator as of the Release 11 shipment date and, if we wish, ship interim updates as ADS programs. Indeed, if we continue to include IGES with AutoCAD at no additional charge, we could simply post the latest ADS executables for IGES for each machine on CompuServe and allow our dealers and customers to download and distribute it at will—it will hardly be of use without a copy of AutoCAD.

The working ADS IGES application would, then, serve as the base for our future ambitious plans for IGES support. Starting with a working, albeit limited, version of IGES will allow piecewise extension and replacement with new, CALS-compliant, code. The new code will be written as ADS-native code, bypassing the emulation library since there would be no need for the new code to be compatible with the conventions of the AutoCAD core. This code will probably use **SGLIB** or an equivalent package for linear algebra rather than the more baroque constructs within AutoCAD—there's no need to make the sacrifices of comprehensibility the AutoCAD routines make for code that's not in the generation pipeline. In time, most of the original IGES code from AutoCAD will have been replaced, and the pieces of the emulation library can be retired as they are no longer referenced.

With Release 12 we could discontinue the built-in IGES commands and support IGES only through the ADS implementation. Alternatively, we could offer Release 11 level IGES support built-in and offer much more comprehensive IGES support through the ADS version as an extra cost option.

Conclusion

"Caustic" has two meanings. In mathematical optics, it refers to the figure formed by rays of light reflected or refracted into a singularity—a mathematical point where the trend reverses, when what was converging diverges, and vice versa. In chemistry, it's a corrosive substance, prone to etch away even substantial structures. As the 1980's pass into the 1990's we seem to stand at a caustic moment in both senses of the word: we're passing through a time when linear projections based on the past provide little guidance for the future; a time when the structures, institutions, and dynamics we've assumed were fixed for many years seem to be disappearing into dust to be replaced by... what?

AutoCAD, as well, is poised at a caustic moment in its history. Having grown comfortably in a fairly linear manner for the last seven years, it is now faced with user requirements and market challenges that dwarf any in its history. AutoCAD will need to grow into something much larger than perhaps any of us can today envision, and must do so on an accelerating time scale. We can achieve these goals: goals I believe essential if Autodesk and AutoCAD are to remain dominant in the market, only if we decouple the evolution of AutoCAD from the necessity of modifying the AutoCAD core code. The AutoCAD core must not just cease to grow—it must shrink in the future. By removing IGES, which in an ideal world would never have been in the core to start with, we can commence this process.

In several years, AutoCAD can come to resemble, not the intimidating monolith of code it now is, but a community of independent and cooperating applications, each developed and maintained by a small team expert in that aspect of the overall task and able to grasp the totality of a program that, by division of labour, has been reduced to a scale the human mind can comprehend. The AutoCAD user will see increased capability, better user interfaces, and improved reliability. If we do our job correctly, the seams won't show, any more than you wonder if the mitochondria in your cells are *really* you. This caustic moment, while marking the high point in the growth of the AutoCAD core, may also be a key milestone on the long road toward AutoCAD's ultimate destiny.

> *. . . all for his sagacity,*
> *Better it is to eschew loquacity.*
> *To make your study grandiloquence,*
> *And busy quibbling devoid of sense,*
> *Argues an empty mind and sick,*
> *In point of fact a lunatic.*

And so, I'll close.

John Walker
Muir Beach, California
December 26, 1989–January 1, 1990
1568 lines of code

The Hardware Lock Strikes Back

I usually don't believe in premonitions. I spent Monday, January 29th, 1990, as I spend most days—sitting at the keyboard working on software development. Usually, I'm as happy as clam while doing this, but that day I felt a most unusual, unfocused mild anxiety: as if something bad were about to happen, but I didn't know what. In the early afternoon, the little flag went up on my electronic mailbox, and, reading the missive that arrived, discovered that Autodesk had decided to re-introduce the hardware lock in domestic AutoCAD Release 11. There are some decisions that are just so dumb I run out of adjectives, and this was one of them. I was also taken aback, given that I was the person who designed the original hardware lock, built the first prototype on my dining room table, ultimately, as president of Autodesk decided to introduce it in international versions of AutoCAD 2.1 and in domestic versions of 2.5, took the heat when U.S. customers arose against us (see page 345), and finally decided, in November of 1986 to remove it (see page 359), that nobody had consulted me or even informed me of this decision to reintroduce the lock. The announcement of the reintroduction of the lock characterised the lock as a "deep emotional issue," as if to characterise those who opposed it as irrational, rather than, based on personal experience, feeling that re-locking the product would not be in the company's best interests.

This is the only instance when, after leaving the management of Autodesk, I drew a line in the sand and said, "I am going to stop this." I immediately posted the following argument against the reintroduction of the hardware lock, and began to organise every resource I could summon to prevent Autodesk from jumping, once again, off the same cliff it had only four years before. I had not only written the copy for a full page advertisement addressed to Autodesk shareholders to run in The Wall Street Journal, I had obtained, from their Palo Alto offices, deadline, submission format, and price information. The next day, January 30th, Al Green reversed the decision and announced that AutoCAD Release 11 domestic would remain unlocked.

First Strike

Date: Mon, 29 Jan 90 13:56:05 PST
From: Ron McElhaney
To: Release 11 Developers
Subject: Meeting on Thursday

As you may know, the company has decided to lock ACAD both internationally and domestically. The h/w

lock which will be used is the Rainbow parallel lock, with which we have had some experience, and which is in wide use by Intergraph on their MicroStation product.

The fact that this is such a deep emotional issue makes it a difficult one to discuss. Most of you will have very strong opinions concerning this decision. Much of that strong feeling is derived from the very painful "exercise-in-futility" which characterized the last attempt to lock AutoCAD a few years ago. In fact, many of you were here at the time, and formed your opinions first-hand. Things are different now, both in terms of the nature of the business itself and in the commercial use of the h/w lock on CAD products, but the emotional content of the issue itself is still high, undiminished by the distance of time.

I have been asked if there are any technical issues which would interfere with our ability to ship a locked product, or which would cause the release of R11 to be significantly delayed. It has been very difficult to separate out the emotional reaction to such a question and to achieve an objective assessment of the effect of this decision on the release of R11. Speaking objectively, the current (although not final) conclusion seems to be that the purely technical issues (which don't require re-working the lock challenge mechanism to make it more secure) are not significant and should not, by themselves, cause a significant delay to R11.

Unfortunately one never deals with purely technical issues in developing software. Getting R11 out on time is a matter of developing software, to be sure, but it is also a matter of deep personal committment to a goal, and a level of self-motivation and hard work which traditional companies very rarely see, but which we see at Autodesk almost all the time. What will the total impact to such a software development organization; an organization which already is working at a level which can only be described as "hyperdrive", and whose pace can be maintained, but only for a carefully-calculated period of time without burning everyone out.

The fact that this decision automatically evokes emotional responses means that is must be widely discussed and explained. Knowing how and why a decision was made, and then disagreeing with it, is much different than merely disagreeing with it based upon an automatic response to it. I have asked Al Green and Malcolm Davies to give a presentation to the R11 developers, for the straightforward purpose of talking about why this decision was made, and how important they feel it is to the success of the company for us to do this. Much of the impetus for this decision arose from the increasing inability of our dealers to be commercially successful representing Autodesk, and Malcolm has invited at least two of those dealers to be present at the meeting to tell their side of this very complex story.

I have asked that the meeting be held on Thursday at 1:00 in the Tech I conference room. Please give very serious thought to the issues which this decision addresses. It is important that you be prepared to discuss your concerns and objections.

Please be there.

—ron

Second Strike

Blows Against The Hardware Lock

by John Walker

Episode II: January 29, 1990

I disagree with the statement that the hardware lock is a "deep emotional issue." I consider the issue of whether to lock or unlock a product to be a straightforward business decision which should be made, like any other

decision, based on the company's overall goals and strategy, and from the best information at hand about all factors involved.

I would, therefore, like to ask the following largely non-technical issues related to the decision to lock the product.

1. How does adoption of the hardware lock benefit our customers?

Autodesk has grown and prospered by always, as much as possible, placing the customer's needs foremost. By customer, I mean the user of AutoCAD, not the reseller, even though we do not sell directly. If we do not satisfy the customer, new customers will not come to Autodesk resellers, but will purchase other products instead. The hardware lock does not benefit the customer in any way of which I am aware.

2. What effect will the hardware lock have on Autodesk's sales and earnings, and on the sales and earnings of our resellers?

It is rare in business to be able to definitively answer a hypothetical question about financial results. One of the few benefits of our Dark Night Of The Soul in 1986 was that we learned the answer: None. Sales did not go up or down when we introduced the lock, and sales did not go up or down when we discontinued it. I know of nobody who predicted this result; certainly I did not. I do not believe that any material changes have occurred in the market since the last time around; in fact, the events since have moved further away from protection devices and schemes.

If the lock is reintroduced with the goal of increasing our sales and earnings and those of our dealers, and consequently improving the business viability of the AutoCAD resellers, I believe that decision to be based on demonstratedly incorrect premises. And if not with those goals in mind, then why?

3. What effect will introducing the lock domestically have on the shipment date of AutoCAD Release 11?

We are presently in the middle of one of the most furious pushes to shipment in the history of the company. Only with Stakhanovite exertion and more than a little luck do we stand a chance of meeting the current release date goal. Already, only for the second time in the company's history, we have disabled features already developed and integrated because we lack the time and manpower to debug and test them as part of the product by the release deadline. Now it is proposed that we throw another major twist into the product cycle. The statement that the lock is already part of the product is abject nonsense; anybody who lived through 1986 will recall that the logistics of acquiring, inventorying, and quality testing locks in quantities adequate for our domestic business involve problems that are not small ones. The vendors involved may be more mature than those we used in 1986, but the volume of locks we will require is also much larger. All the issues of documentation, installation problems, compatibility with hardware, and the like are identical. We introduced the lock in the development cycle of AutoCAD 2.1, which was, before Release 11, our most critical date-driven release (being scheduled near the time of the initial public offering). I think it is more than coincidental that AutoCAD 2.1 was the worst release of AutoCAD we have ever shipped, with diversion of company resources into lock-related issues a major contributor, if not the proximate cause.

4. Who are the additional customers who will buy a locked product?

Introducing the lock increases our cost of goods (perhaps doubling it, if some of the numbers I've heard bandied around are to be believed). Therefore, if margins are not to fall, additional sales must be generated (indeed, as noted in item 2 above, unless the lock is a public relations exercise or a moral crusade, this is the only

reason for considering it). Therefore, what is the profile of the customer who will buy a new AutoCAD from an AutoCAD dealer if the product is locked, but would steal the product were it not? People tell me I have a fairly vivid imagination, but I cannot come up with a sketch of a sufficiently large population of customers representing that foregone revenue.

Remember that shortly after the lock is reintroduced, products will appear that circumvent it. (The suggested lock, and its implementation within AutoCAD, will be much easier to defeat than the lock of 1986). If our 1986 experience is a guide, the market price of the lock-defeating programs will be less than $100. In that environment, the question becomes this: "Who is there who today would steal our product, but who will pay $2000+ for a legal copy of AutoCAD rather than purchase a $100 program that lets him continue to steal it?" Legal remedies against lock-defeating programs are probably impossible and ineffective in any case.

I would suggest that those advocating the lock without pondering the full implications of this issue (which I did not appreciate until tutored by brutal experience) are victims of the same kind of naive reasoning that suggests that if one doubles taxes, tax receipts will also double. The real world is a complex web of nonlinear relations and interconnected feedback loops with delay. When you change a parameter, you're shifting incentives, not results, and you have to think out all the consequences, not just the obvious first-order ones.

5. What does the hardware lock decision say about the direction of the company?

This is not just an issue important to those who presently work for Autodesk; it is central to the overall mission and strategy of the company, its relationships with its resellers and their customers, with the marketplace and our competitors, and with the perception of Autodesk among all these constituencies.

Autodesk has always adopted a strategy of broad market share, low price, rapid enhancement, and responsiveness to the user. Almost since inception, AutoCAD has been the "safe buy" because so many other people use it, even if some have not yet paid. This strategy has served us well, and I am absolutely convinced that, at least in the domestic market, rampant piracy has substantially contributed to our current dominance of the market. I wish there were a way to measure the number of current legal copies of AutoCAD that replaced pirated copies. It would not startle me to discover that number to be very large.

Reintroducing the lock sends a message that Autodesk has changed its strategy. I'm concerned here with the message, not the strategy itself, and I consider irrelevant on this point comments not grounded in the 1986 experience. Regardless of the moral and intellectual merits of the arguments employed, the simple fact was that the lock was perceived as Autodesk abandoning "the little guy" responsible for its initial success (and I got at least as much of this from dealers, who we were trying to benefit, as from users).

This comes at a time when Autodesk is sending many other signals that seemingly herald such a change in course. We have opened regional offices. We have announced an aggressive discount program for the Fortune 500 and the government (and if you want to get onto morals, I think the concept of selling to General Motors for less than the price paid by a one-man consulting firm is as least as reprehensible as profiting by software piracy, especially when practiced by a supposedly entrepreneurial upstart company). We have concentrated development on network licenses, again aimed at the larger customers.

It is worth reflecting on the fact that despite all the hoo-rah about Fortune 500 and government sales, they're still about 15% of our business (and presumably a much smaller component of our typical reseller's business). Sending the wrong message to the people that are responsible for 85% of our revenue can be disastrous. I saw the unanticipated misperception of our intent in 1986, and I believe nothing has changed. If anything, the scars of that experience have hypersensitised our users to the issue of locks.

6. What will be the effect of the lock on the perception and initial acceptance of Release 11?

Version 2.1 was one of the most significant product introductions in the company's history. With the addition of AutoLisp, AutoCAD set itself on the course that has brought us all here today. (Full AutoLisp was deferred until 2.18, an update release, because of time-driven pressure to ship... sound familiar?) Yet the major enhancements in the product were simply lost in the furor surrounding the lock. In fact, virtually nobody paid attention to the capabilities of the product at all; it was just locks, locks, and more locks.

Now if we're pushing Release 11 because we believe both that the features it contains will increase sales and that the mere fact of a new release contributes to revenue, then can we afford to have the normal publicity attendant to that release consumed by a second hardware lock firestorm?

I urge you to think carefully about this issue. If Release 11 is so critical as to justify the efforts and compromises attendant to its push to shipment, the risk of a public relations disaster coincident with shipment is grave indeed. Having presided over the last one, I consider anybody who minimises the risk of repetition as uninformed, misguided, or naive. It may not happen, but what are the probabilities? How did you arrive at your estimates? What happens if you're wrong?

In addition, the 1986 experience demonstrated that Autodesk will cave in on locks when an uproar arises. (The argument that only a small percentage of our current users were users in 1986 is fallacious; information about the 1986 episode will be spread through all available channels once the lock surfaces. It's in The Autodesk File, after all). Consequently, there are strong incentives to:

a) Make a big stink about the lock, because that made us remove it the last time, and,

b) Defer buying Release 11 both to "send-em-a-message" and because one feels the probability is high that one will be able to buy (or steal) an unlocked version within a few months anyway, once Autodesk concedes defeat this time around.

Are these the incentives we want, or need, associated with Release 11's introduction?

Now these are just the obvious issues related to the domestic reintroduction of the hardware lock—a compilation off the top of my head based on the 1986 experience. I have not thought through the numerous, more subtle, consequences that appeared in 1986 in terms of the current proposal. I presume that in the process of arriving at the decision to reintroduce the lock, management considered all these points and more (and if not, then the decision was arrived at through a dangerously flawed process). I am therefore interested in the conclusions that were reached in regard to each of the issues I raise herein, and the background information and reasoning used to arrive at them.

If any emotion is involved in the issue of the lock, it is emotion that springs from seeing our company about to repeat the single worst experience in its entire history, with a probability of disaster I would estimate on the order of 85%. It is emotion born of seeing a company forget its past, its goals, its customers, and the principles and processes that made it so successful in the first place.

Escalation: Eyeball-to-Eyeball

Date: Tue, 30 Jan 90 15:40:29 PST

From: John Walker
To: Al Green
Subject: Letter re: hardware locks
Cc: Malcolm Davies, Dan Drake, Greg Lutz, Ron McElhaney

As a selling shareholder in the S-3 offering of June 20th, 1989, I remain personally liable until June 21, 1990 in shareholder suits based upon inadequate disclosures in the prospectus.

Should the hardware lock suggestion go into effect and cause a catastrophic drop in the stock price, shareholder suits are a virtual certainty.

To those unaware of the nature of my current day-to-day activities, and know of my personal involvement in the last hardware lock episode, documented in "The Autodesk File" for all to read, any claim that I had nothing to do with the decision to re-lock the product would probably be given the same credibility as Mr. Reagan's claim not to remember anything between 1981 and 1989.

Consequently, I would like a letter drafted and signed by you, then notarised, that states the following:

> John Walker was not consulted in any manner regarding the decision to reimpose a hardware protection device on domestic copies of AutoCAD.
>
> John Walker did not participate in any form in the deliberations prior to this decision, nor did he participate in the decision itself.
>
> John Walker was not informed of the decision to reimpose a hardware protection device on domestic copies of AutoCAD before 1:56 P.M. PST on Monday, January 29, 1990, at the same time all other members of the AutoCAD software development group were informed of this decision.
>
> In addition, John Walker has attended no senior management, board of directors, or equivalent meetings since June 1, 1989, and has not otherwise received the information presented at those meetings, nor participated in any decisions made there.

Capitulation

January 30, 1990

The decision to reintroduce the hardware lock was not made without a great deal of time, thought, and heartache by management. I was on the front line of the last firestorm and am not anxious to hurl myself headlong into the snakepit again without good reason. O.K., now that I've said that, the reality is that this issue has, to say the least, attracted quite a lot of attention. The good news is that it's refreshing to know that the people at Autodesk still care extremely about the direction of the Company. The bad news is that already a great amount of time, thought and anguish are being given to one topic at a time when we are under the greatest pressure that we've ever had to get out an AutoCAD release. The hardware lock is not going to drive a wedge through the Company and I'm not going to risk the release of Version 11 through stubbornness. The lock will not go on Release 11 and the Thursday meeting to debate the issue is cancelled. Let's direct all of our energies to getting Release 11 out on time.

Alvar Green

“Autodesk Arrogance”

Three years had passed since I cut loose with my last Jeremiad at a company meeting (see page 385 for the last one). Autodesk had continued to prosper in those years and, with growth, it’s inevitable that many new people arrive at the company who haven’t shared in the company’s evolution and perhaps don’t fully understand, therefore, the foundations of the company’s success.

Well before the clash over the attempt to reintroduce the hardware lock in early 1990 (see page 524), I had been picking up more and more signals that indicated that Autodesk was increasingly adopting a “big CAD company” outlook and distancing itself from the small customers who continued to account for the overwhelming percentage of our sales. The word “arrogance” kept cropping up; both from the dealer and customer community, applied to Autodesk’s behaviour, and within the company, referring to old-timers who thought they knew better ways to compete in the CAD business than the way other CAD companies were operating.

I decided to grab a cinder block and take this head-on. At the company meetings of February and March 1990, I tried to bring the focus back to the customers whose purchase decisions were the underpinnings of Autodesk’s success. February was the gentle introduction, March the gnarly climax.

Remarks for the February Company Meeting

by John Walker

February 9th, 1990

The end of January marks the close of Autodesk’s financial year. It’s a time devoted literally to summing up the results of the last twelve months’ successes and failures, where the brilliant breakthroughs are added and bonehead blunders subtracted to yield a number that tells us all how we’re doing.

The end of January is also the anniversary of the meeting, now eight years ago, in my living room where the idea of starting this company first took shape.

So about this time every year, I find myself looking back over this weird adventure we’ve lived through and thinking about what made it all possible.

For all of our success, our salaries, our bonuses, our recognition in the industry, our high valuation in the stock market, our exciting new technological ventures, our pioneering of new markets around the world, our

innovative programs in education, our investments in emerging industries all stem from one common source.

Everything we've done, all we've achieved, and all we hope to do in the future is made possible by the customer who buys Autodesk products.

When you see those numbers with lots of zeroes to the left of the decimal point, it's all too easy to forget that they're nothing but the sum of a lot of individual little numbers. What is the reality behind those numbers?

Just this. The fundamental event that makes everything we're doing possible is a customer going to an Autodesk dealer, parting with somewhere between two and three thousand dollars, and walking away with a copy of AutoCAD.[311] Who are these customers? Well, for all you hear up here month after month about Fortune 500 and the government, about 85% of them are individuals and small businesses. This shouldn't be surprising when you consider that all the creativity and vitality in the economy, however measured, is among the customers that overwhelmingly constitute our market.

Since only a small percentage of you here have ever *bought* a copy of AutoCAD, I want you think about what that involves. You are exchanging around three thousand of your dollars for a cardboard box filled with paper and rusty plastic that promises, after you've mastered its self-evident complexities, to repay your investment in it.

Now some deep-thinking analysts may tell you that customers are buying CAD out of fear of their competitors, because they want to appear technologically current, or other subtle and indirect reasons. What a pile of crap. I was a small businessman, and I can tell you then whenever I peeled a hundred and fifty twenty dollar bills off my anorexic wad to buy something, it wasn't without a lot of thought leading to a firm conclusion that it was worth it.

That's why our customers pay what they do for AutoCAD. Because it's worth it.

If we lose sight of this simple fact, and veer off into directions and priorities that do not put value and service to the customer at the forefront of everything we do, all this will end. We will fail, and we will richly deserve to.

If we continue, as we have done consistently for the last eight years to measure every proposal against the standard, "How does this benefit the customer?",[312] I believe the success we've experienced to date will be just the base upon which far greater achievements can be built.

The confluence of developments in several key technologies suggest that the industry in which Autodesk is now the predominant worldwide force will be at the center of a revolution in manufacturing more profound than the introduction of steam power that heralded the industrial revolution, or of silicon technology that ushered in the present information age.[313] If we position ourselves properly for the discontinuous changes that I now believe are likely, I can envision no limits to our success in the era that is almost upon us.

But to achieve anything at all, even our next quarterly numbers, we must never forget our customers. It is the customer, ultimately, that we are working for, and it is the customer who we must always strive to satisfy. All the rest will take care of itself, in the fullness of time.

[311] The section on "Quantum Economics" in *The New Technological Corporation* (see page 469) explains this simple concept in a page of tangled text.

[312] Compare the "benefit the customer" discussion in regard to the hardware lock on page 526.

[313] This is a cagey reference to nanotechnology. I was just now starting to work on a Tech Forum presentation on nanotechnology and how Autodesk might participate in its development, which was eventually presented in May 1990 (see page 558).

Remarks for the March Company Meeting

by John Walker

March 2nd, 1990

These meetings have a tendency to degenerate from the kind of open forum for airing important issues that affect our company into a kind of rah-rah, go-team pep rally. Hey, even in high school, the cheering and the bonfire were just the price you paid to get to the beer. So before everybody goes to sleep, I'd like to inject some controversy into this happy assemblage.

I think we need it. Badly.

I want to talk about arrogance. I will be brief.

Recently, several people have approached me and reproached me about "Autodesk Arrogance". By this, they don't mean the self-destructive kind of arrogance where a vendor becomes insensitive to the needs of its customers and dealers—the arrogance that has led to the downfall of many a high-flying company—they mean something entirely different.

I have heard the term "Autodesk Arrogance" used to denigrate those who believe, as I do, that the practices and principles that built our company from a hard-scrabble start-up into an industry leader should continue to govern our growth from today into the foreseeable future.

I have heard the term "Autodesk Arrogance" applied to dismiss those who believe that continuing to focus on the most productive sector of the global economy, the individual and the small company, will continue to serve Autodesk as it has in the last 8 years.

If betting on the creative individual, against such brain-dead dinosaurs as General Electric, McDonnell Douglas, Ford Motor Company, and the United States Government is arrogant, then call me arrogant.

But what, indeed, *is* arrogant?

Is it not arrogant for people in an office building in Marin County, California to assume they know the needs of our customers better than the customers themselves? I think that's arrogant, and foolish.

Is it not arrogant to engage in so-called strategy based on five-year projections of the market when not a single so-called market analyst in 1982 predicted the dominance of AutoCAD a mere five years later? I think that's arrogant, and dumb.

Is it not arrogant for so-called strategists to decide what the market is "ready for" and "not ready for", when history has repeatedly shown the market and the customers who compose it to be much wiser and more resourceful than any self-appointed analyst? I think that's arrogant, and shameful.

Is it not arrogant to insult any customer with the notion that he will choose a product for any reason other than a carefully arrived at and rational judgement that it's the best? I think that's arrogant, and dangerous.

Is it not arrogant to withhold tools, products, features, and information from our users, our dealers, our developers, and all the other members of the community assembled around our products because of a presumption that "they won't understand", "they aren't smart enough", or "it'll only confuse them"? I think that's arrogant, and counterproductive.

These are the kinds of real arrogance I think we need to guard against. They take root all too easily and quickly grow to the point where we become distant from the people responsible for our success: our customers and the dealers who sell them our products.

Now some people may hear this and say, “Right on! It’s about time somebody told the truth about that other department!” Look closer to your own desk first. These problems are everywhere in our company.

The kinds of arrogance I mentioned are deeply ingrained in the business culture of the United States, particularly in the large companies. To a certain degree, they are taught in many business school curricula. People who come to Autodesk from other companies are usually aware that Autodesk is not a normal company. Normal companies aren’t this successful. Ever wonder why?

I think it’s this. We constantly strive to make the best products possible, to effectively communicate information about them to a wide variety of people, to make our products broadly available, to expand the market by delivering ever growing capabilities at prices people can afford, and to ultimately trust the judgement of the market to tell us what we should and shouldn’t do—to guide the evolution of our products and our company.

Most places, you come up with an idea and the boss says, although not usually this directly or honestly, “I don’t care what you think. What matters is what *I* think.” That’s arrogant.

Around here, I’ve been known to say things like, “I don’t care what you think. What do the *customers* think?”. That may sound arrogant, but to me it’s just plain old common sense. The evidence that it works is all around us.

“But that isn’t how they did it where I used to work!” Well if that place was more successful, more fun, and more rewarding than Autodesk, why are you here?

“Growth has to change how you do things. Large companies can’t remain responsive.” Frequently they don’t. Then we drive them out of business.

If refusing to discard a formula for success that built our company from nothing to a billion dollars market value constitutes arrogance, then call me arrogant.

I will wear it as a badge of honour.

Why Molecular Modeling?

In early 1990, Autodesk was evaluating HyperChem, a Microsoft Windows-based molecular modeling package developed by Hypercube, Inc. (Details on how Autodesk and Hypercube were introduced may be found on page 558.) Technical evaluation of the product quickly established that the product was scientifically correct, well-built, and provided unprecedented ease-of-use for a molecular modeler. The biggest question in the minds of Autodesk management was whether the molecular modeling market was big enough to justify the effort, and if it made sense to get involved in a product so technically complicated, computationally intense, and specialised. The following is an advocacy piece I wrote which argues that every single one of the present doubts about HyperChem could have been raised in 1982 against AutoCAD. On August 17, 1990 Autodesk made an investment in Hypercube and obtained exclusive distribution rights to HyperChem; the product shipped on March 30, 1992. In January 1994, the Scientific Modeling Division was disbanded and the distribution of the product terminated.

Why Molecular Modeling?

by John Walker
March 7th, 1990

As we proceed with evaluation of a possible business relationship with Hypercube, I'd like to briefly discuss the most fundamental question, "Why should Autodesk get involved in molecular modeling?" Agreement with the reasons I'll describe below and with the conclusion that we should enter that market when an attractive opportunity to do so appears does *not* necessarily mean we should conclude a relationship with Hypercube; that specific decision must hinge on our evaluation of that company and its principals, products, and strategy. But conversely, if you conclude that molecular modeling is a hopelessly narrow, arcane, and hard-to-sell niche market, there's no reason to waste any time evaluating the particulars of Hypercube.

One of the major tasks I had set for myself in the early part of this year was to write a paper that laid out my view of the currently emerging technologies in which Autodesk could become involved with relatively little investment or risk, and which were consistent with the overall direction of the company and the businesses in which we do well. I had scheduled a Technology Forum for January 27th titled, "What Next: The Coming Revolution In Manufacturing" to discuss just these items.[314] Unfortunately, the recent spate of alarums and diversions has pushed that task further and further back in the queue, necessitating the somewhat sketchy arguments presented below. In any case, things are happening in this area much, much faster than I or many

[314] This talk was finally given on May 10, 1990. A transcript appears starting on page 558.

of the most voluble visionaries in the field had expected only a few months ago. I have been accumulating a file of clippings to circulate with the paper, should I ever finish it, and it seems like every week's *Science* and *Nature* contain increasing quantities of relevant material.

I believe that molecular modeling passes the two key tests I look for in evaluating any potential new product area. These filters are the same I applied when screening initial product ideas for Autodesk, and the fact that AutoCAD met these criteria was one of the reasons I strongly supported it as a product for Autodesk.

Filter 1: It stands to benefit from increasing compute power

The safest bet in the world is that computing power will continue to grow at an exponential rate while costs stay constant or fall. There is simply no technological barrier on the horizon to stop this process, and market forces are inexorably driving the evolution of technology in this direction. The best way for a software vendor to profit from this is to choose a problem domain that has demonstrated its worth on large computer systems, then repackage that tool for a much broader market, anticipating the arrival of low-cost, mass-market computers that will allow it to be used practically.

This is precisely what we did with AutoCAD, and we knew exactly what we were doing at the time. By choosing a product on the edge of currently available compute power, you avoid jumping into a heavily contested market. Instead, you can start the process of market development on the margin, among education and other users able to use the product on existing machines, putting the infrastructure in place so that when machines really suited to the product begin to arrive you're in an unassailable position of strength.

Actually, with AutoCAD, we thought the PC/XT was the machine that would make CAD real on the desktop but we were wrong. Looking back on what really happened, the whole XT era was precisely the time of market-building I described above. When the AT came out, serious practical work with desktop CAD was truly possible, and we simply rode the wave we were already perfectly positioned on.

In looking forward and choosing new products, we should be seeking products with the same properties; products which run on large computers which are useful enough that a small segment of the market able to afford such machines pays the price of admission, but which will be applicable to a much broader set of users when the price falls to a level they can pay. This is a property of many of Autodesk's current development directions: photorealistic rendering, solid modeling, finite element analysis, and 3D user interaction.[315] Molecular modeling has the capacity, like CAD, to soak up all the additional computing power anticipated for the next decade, and in the process, expand the market for computational chemistry just as AutoCAD has done for CAD.

Filter 2: It is central to emerging technologies

While one can unquestionably expand a market by simply lowering the price, ideally we should look for products in markets which themselves are expected to grow rapidly. CAD and multi-media are two such markets in which we're currently positioned, and I believe that molecular modeling is one of the key enabling technologies of events about to unfold which may dwarf either of those markets. CAD, for example, is the key enabling technology for most of the changes underway in manufacturing and automation. It stands to reason that as

[315] A.k.a. virtual reality or "cyberspace." See page 439.

those technologies expand from their initial small and expensive base, CAD will penetrate a much wider market. Indeed, this has happened. I think that much the same is true of molecular modeling.

First, let's step back from this "computational chemistry" stuff (which can't help but conjure up the thought of strange and pungent odours emerging from your floppy disc slot) and recall that molecular modeling is really nothing but mechanical CAD at the atomic level. What you do with such a system is assemble a part atom by atom, using all the normal CAD-like commands, view the model, then perform various kinds of analysis upon the model to study and predict its behaviour.

Working with atoms and quantum mechanics is a whole lot different than machined metal and Newton's laws, but what you're doing in the overall design cycle with such a system isn't all that different from the way an MCAE user exploits solid modeling, FEM, and postprocessing.

If molecular modeling seems intractably difficult computationally and arcane in its terminology, much the same could have been said 10 or 15 years ago about finite element analysis of nonlinear materials with integrated computation of mechanical and thermal properties. Today, that technology is being used to design tires for your car.

The reason we should be interested in "atomic CAD" is that, sooner or later, it's going to be the most explosively growing segment of the CAD business. Ever since the 1950's we've been making stuff smaller and smaller. Eventually this will end; when you hit the level of atoms, you can't go any further (this assertion may seem glib, but the argument in favour of it is very strong, but irrelevant to this discussion). But when you hit the atomic level, physics, chemistry, and engineering become unified into a single coherent field yet to be named "Molecular Engineering." This will simply sweep chemistry, the study of atomic interactions, into the unification of solid state physics and engineering that has already occurred in the development of semiconductor device technology.

In talking about "hitting the level of atoms" I am not envisioning some far-out event in the mists of the twenty-first century. Several current technological developments such as the scanning tunneling microscope, the atomic force microscope, quantum well transistors, molecular optical memories, and protein engineering, all subjects of well-funded and aggressive research programs at places like IBM, DuPont, Bell Labs, and Texas Instruments, are already reaching this level. The journal *Science* inaugurated their new series of survey articles, "Science in the 1990s" with a review of atomic-level technologies and predicted the consequences would be at the heart of many scientific and technological developments in the next 10 years. When this really begins to roll, being positioned as a leader in atomic level CAD is going to be worth a great deal, indeed.[316]

But there's no need to lose money until this happens, or even if it never does. The beauty of molecular modeling is that it is already a viable business, one in which we can apply all the same strategies of penetrating education, broadening the market, and riding the curve of increased hardware power just as we did with AutoCAD. Every step we take, and every success we have in this developmental period will just put us in a better position for the time when molecular engineering explodes into exponential growth in the manner integrated circuits and microprocessors did in the 1970s and 1980s. And if it never happens, then we'll still be at the centre of the rapidly growing biotechnology market, where molecular modeling is already essential.

[316] I studiously avoided using the word "nanotechnology" in this paper because it had already begun to accrete an odour of flakiness, due to the apparently outlandish claims made by some of its partisans. In the Tech Forum presentation the following May (see page 558), I used the word only once, as a synonym for my preferred term, "molecular engineering." Since 1990 the word "nanotechnology" has become more respectable, and is now seen as an entirely legitimate area of research.

Conclusion

I believe molecular modeling represents a major opportunity for Autodesk. It seems to share many of the same properties that contributed to the success of AutoCAD, and to be at a stage of market development similar to desktop CAD at the time of AutoCAD's introduction. Whether the present opportunity proves attractive under closer scrutiny or not, we should continue to seek ways to position ourselves in this market.

Whither AutoCAD?

Ever since AutoCAD began to include 3D and other facilities oriented more toward model building than production drafting, there had been an ongoing low-level controversy over how Autodesk should proceed in broadening its coverage of the CAD/CAM market: develop new products with links to AutoCAD, or simply grow AutoCAD into an integrated tool which encompassed high-end modeling as well as drafting and detailing?

This paper, written in early 1990 by then-V.P. of Technology Ron McElhaney, describes the CAD/CAM market, AutoCAD's place within it, the difference beween Autodesk and the traditional high-end vendors, and the opportunities in the newly emerging field of Mechanical Design Automation. Ron suggests a course of action to allow Autodesk to position a new product in this domain just as AutoCAD had succeeded in the drafting sector.

Whither AutoCAD

by Ron McElhaney
March 15, 1990

AutoCAD has a leadership position in the desktop CAD software market. The reasons for this are many, but include the fact that Autodesk sells useful products to a customer base whose needs it fully understands, and with whom it has built a deep and lasting relationship. We are now beginning to see increased competition from the "high-end" turnkey vendors. In fact, we are now seeing them begin to adopt some of our own successful business practices. The use of dealers, the stress of open architecture, and the encouragement of third-party developers are serious competitive responses which should convince us that these vendors understand the importance of beating us in the open market. What must Autodesk do to guarantee that it will be Autodesk which survives the inevitable high-speed collision, and not Intergraph or Prime? More importantly, if we succeed in displacing companies such as Intergraph, will we do it at the cost of estranging those very customers who have been responsible for our great success?

Introduction

The battle lines in the war for CAD dominance are just beginning to form Initially almost unnoticed in its rapid rise to success by the "high-end" vendors, Autodesk is just now beginning to skirmish with full-function, turnkey-vendors such as Intergraph. Short-sightedness (or more properly, blinding stupidity) on the part of

Intergraph management has led them to believe that we have been isolated in the "low-end" category of CAD/CAM, and to design an attack on us there.

The guns they are bringing to bear on us are aboard their MicroStation product, their desktop solution which they offer in direct competition to AutoCAD, while their flagship product aims its cannons at more 'sophisticated" competition.

What Intergraph fails to realize, of course, is that by combining more and more functionality with greater and greater openness and adaptability, we at Autodesk will soon have the ability to create whatever we need to provide the customer useful products at all levels, from high to low and everything in between, eventually to include a battleship that will blow them out of the water.

The thought of breaking through the barricades, confronting the enemy on his own ground and defeating him with superior weapons is almost poetic in its justice and simplicity.

Autodesk as High-End Vendor

The opening argument has all the right elements for being the call-to-arms for Autodesk development. It is aggressive, suggesting at a violent confrontation with our competitors on the high end, with a victory for Autodesk guaranteed after the battle is done. It discredits their way of life and promises a drastic, but deserved, fate for the turnkey vendors, a result consistent historically with what Autodesk has done to its competition in desktop CAD.

Not surprisingly, this argument seems to be accepted within Autodesk almost without argument, representing, in some sense, the default position concerning the future of AutoCAD in the CAD/CAM industry. It takes for granted that the only differences between the turnkey world and the world of desktop CAD are technological. It assumes that as more and more power accrues to the user of desktop CAD, and as we at Autodesk begin to add more and more technology to our system, the differences between turnkey CAD and desktop CAD, both perceived and real, will diminish, resulting in desktop systems offering essentially everything which the turnkey vendors can, but doing it in a more cost-effective way.

A simple-minded extrapolation of AutoCAD, based on our continually increasing its technical content, appears to cause it, at some point in the not-too-distant future, to be positioned directly against the high-end vendors. This simple extrapolation hints that history may repeat itself, causing a future AutoCAD, chock full of powerful technology, to become the product of choice for all CAD users, as they leave the turnkey vendors *en masse* for a cheaper, better solution.

This argument is exactly what investment analysts and industry consultants want to hear. It promises increasing market share, continued quarterly dividends, greater and greater margins, and an earnings per share which increase off the charts. It is music to their ears, and they react in predictable ways, helping to send our stock higher and higher. That's where the real action is, at the high end of CAD/CAM, isn't it? Is it?

The Business of Autodesk

There are several serious problems with this simplistic view of AutoCAD's future. First, although the most obvious differences between the desktop vendors and the turnkey vendors are technological, the most significant differences have very little to do with technical content. The most important differences have to do with basic philosophical issues.

AutoCAD is undeniably the most popular CAD product in the world. We all know the impressive numbers: several hundred thousand licensed users, and many times that in illegal copies make AutoCAD the practical standard in desktop CAD everywhere CAD is done.

How has Autodesk managed to be so successful? We can all count the ways, but when we do, we find that providing superior technology is not one of them. In retrospect, Autodesk succeeds because it is a superior business.

Dedication to the customer and serving their needs has turned out to be more important than providing leading-edge technology. That attitude of presenting a strong, almost fanatical dedication to the user is apparent in everything we do, and is recognized by the user who rewarded us in spades.

The "bulking up" of AutoCAD may not guarantee us any greater success than we now have. it may, in fact, achieve exactly the opposite, especially if high-end features are arbitrarily added to the system without regard to the benefits which they bring. It is clear that much of the current customer base of AutoCAD has little need for many of the advanced features which we are contemplating adding to future versions of AutoCAD.

As we add NURBS-based solids modelers, constraint management systems, and other "sophisticated" features, we should ask ourselves what our users, who have purchased AutoCAD for years to do their drafting/drawing work, will do with these fancy features, Will these users see each additional release of the product promising them only unneeded features and increasing complexity? If the added value is not perceived to be great enough, the simplest decision users can make is to buy a system which better serves their needs.

The potential effect on our business would not be confined solely to those users which were left behind in our "dust" as we climb the technology curve. By orienting our marketing to the customer base which *does* have a need for these features, we will be forced to position ourselves in a market with which we have a small collective experience, and where our ability to succeed is untested. Can we risk the abandonment of a large number of loyal customers who have helped us to achieve our current enviable position of market domination and financial success?

I do not believe that one product can span the range from desktop CAD through high-end CAD, and that for AutoCAD to continue its world-beating success path, it must continue to provide solutions which are consistent with our current users' requirements and with our own experience. I feel strongly that adding technology which will not contribute to making AutoCAD a better and better drafting and drawing system would be a serious, perhaps fatal, mistake. This must continue to be the most important goal of the company.

An Opportunity

At the same time, we cannot ignore other technologies such as solids modeling and parametric design. A large amount of money is spent each year on 'high-end" technology. As I will discuss later, most of this is spent by

large companies in the process of making high-level, corporate-wide strategic decisions, decisions which have very little to do with practical, day-to-day efforts, but with systems and technology which, it is believed, will affect the company's future, and its ability to compete.

I believe that a real opportunity exists in the area of what is being called "design automation." It is here that the next attack on the turnkey vendors will take place and, with Autodesk's participation, this attack may be fatal. The last large attack was, of course, mounted by companies such as Autodesk, which drew large numbers of users away from the turnkey vendors, and which left them whole, but broken.

Within the vanguard of the movement toward design automation were companies such as Cognition and Aries who, like many companies first on the scene, could not see the future clearly enough to compete successfully against the turnkey vendors. The second wave is represented by companies such as Parametric Technologies Corporation and Ashlar Vellum, and many others sure to follow. These products now offer clever and innovative approaches to design, and will surely contribute toward changing the methods of design engineering in much the same way that the Macintosh changed the way we interact with computers.

Autodesk should be in this fight for dominance in the engineering design market. The most important reason is that this market provides us with an opportunity to take advantage of the weakness and stupidity of our high-end competition; and gives us a real chance to take a run at leadership of the traditional CAD/CAM/CAE market by offering a useful, well-crafted, competitive system at the so-called "high-end" of design. This should not be difficult for us to do and the probability of success, in some significant form, seems to be almost certain.

Secondly, we have all the technology necessary to be competitive in this market, plus an advantage shared by no one else: an awesome presence in the CAD/CAM industry throughout the world, and a loyal following which is ready-made to extend our mastery from drafting/drawing to engineering design.

It will not simply be a matter of technology which will determine our success here. It is my view that the two industries of desktop CAD and turnkey CAD/CAM/CAE are vastly different, one from the other, and a strategy of success in one market will not guarantee similar success in the other. Not only are the companies which provide product for these two markets themselves different, but the motivation which drives them to develop features for their customers is also different. In fact, the users of each type of system themselves share few similarities, especially in what they expect from their systems.

For this reason, and many others, I do not believe that we can be successful attempting to cover the broad market, providing solutions for "low-end" drafting and drawing, and high-end design and integration with one product, and that a successful strategy will require covering the market with different products.

This strategy would have us create two products. One of them would remain AutoCAD, allowing us to continue our self-evident path of success in the area of drafting and drawing production at which we are the world's most successful.

The second product, based upon AutoCAD, could be aimed, easily and almost without effort, toward filling a vacuum at the high-end, left by the short-sighted, old-fashioned, and bankrupt practices of the turnkey vendors. Such a product might be called "AutoCAD Designer."

Containing competitive technology, and focused on solving important problems in the design segment of the manufacturing process, as AutoCAD was focused on the drafting/drawing process, the Designer would provide Autodesk with the ability to make significant inroads into a customer community which has traditionally been all turnkey-based.

If we were to achieve success here, there would be few places left for these dinosaurs to hide and a drastic "re-adjustment" should occur in the high-end vendor community within a few years as the very foundation of their business begins to disappear. Almost as significantly, we would be able to short-circuit the success of companies such as Parametric Technologies Corporation (PTC), whose customers would find it an easy decision to transfer their loyalty, and their money, to a company which is becoming an institution in the CAD/CAM market.

The High End: What is it?

I mentioned before that these two industries are different. In fact, I believe that they are, in the final analysis, more different than they are similar. An understanding of these differences will lead to a deeper appreciation of what approach we should take to displacing the turnkey vendors. In the following few pages, I will provide an in-depth discussion of the world of high-end CAD/CAM. My purpose is not to honor it, but to show how different are the fundamental factors which influence it, both from within the turnkey companies themselves, and the companies and individuals which buy and use

When l refer to "desktop CAD," I am of course referring to products such as AutoCAD; products which were first offered on desktop machines, and which gradually evolved into what they have become today. I am not referring to transplanted technology.

By turnkey CAD, I refer to products variously termed 'high-end" or "full-function" CAD/CAM/CAE/CIM/... products which contain a wide range of applications bundled within them, often packaged with a workstation and priced as a single unit. These are generally still big ticket items, and the companies which produce them are limited in their business outlook by the "traditional" view of CAD/CAM.

Traditional CAD/CAM

CAD/CAM as a sophisticated industry actually got its start within large industries which had the most to gain from automating the design and manufacturing process. Historically, these large companies, especially those within the aerospace and automotive industries, have been responsible for some of the most significant advances in the art and practice of CAD/CAM.

In the late 1960's and early 1970's entrepreneurs began to create companies which were based upon this emerging technology, many of them still in existence in 1990. The turnkey industry is, of course, much older than the desktop CAD industry, but it is also much older than the entire PC software industry. For example, Calma began business in 1968, Computervision, Applicon and Intergraph (as M&S Computing) in 1969.

During that period of more than twenty years, the turnkey vendors have managed to develop some very useful applications based upon sophisticated software components, and have changed hardware platforms (and even languages) many times. By the time desktop CAD appeared on the scene in the early 1980's, the turnkey vendors had already invested thousands and thousands of man-years in the development of their systems and had solved most of the basic problems associated with design, drafting, analysis, and manufacturing.

Although the companies representing the turnkey CAD/CAM industry are not doing particularly well lately, they have as a group introduced a generation of engineers and designers to the use of computer-assisted techniques,

and in their efforts laid the foundation for the eventual success of desktop CAD.

One of the areas in which the turnkey vendors have made significant progress, far beyond the ability of most manufacturing companies to keep up through internal development, is in their ability to represent and manipulate "difficult" geometry. It is in this area that the level of technology between early desktop CAD and turnkey CAD differed the most. Whereas desktop CAD provided lines, arcs, circles, and occasionally splines and conic sections, the turnkey systems offered a fully parametric wireframe and surface modeling capability attempting to support sophisticated mechanical applications such as 3- and 5-axis N/C. This difference in modeling representation is only now beginning to change as the desktop vendors begin to climb the technology curve.

What caused the rapid success of desktop CAD, and why didn't the turnkey vendors themselves benefit by the revolution which was plainly occurring? Why wasn't Computervision or Calma, for ex- ample, the first one to offer CAD on a desktop? Why did the success of desktop CAD await the founding of companies such as Autodesk?

Although these questions are fundamental, the answers to them have more to do with self-image and perceived destiny, than with business.

To begin to understand how truly different Autodesk is from the turnkey companies, consider how these high-end vendors view themselves.

The View from the Top

One of the reasons that Autodesk was ignored by the turnkey vendors in the early days of desktop CAD was that the developers of the high-end systems honestly discounted any value which desktop CAD systems would be able to offer to users. The concept of "CAD-on-a-PC" directly conflicted with their under- standing of their own essential purpose in life, which was to bring order, power, and complete integration to the design and manufacturing process.

A CAD/CAM veteran at that time would have described his system as something to design with. "Sure," he might have said, "our customers use our system to do production drafting, but its true value derives from the fact that these users can create a complete digital representation of their models, and all the integrated applications which we offer can access this model directly, taking whatever application-specific data it needs. Thus there need be only one copy of the data for everyone." He would have smiled smugly at this point.

"FEM users can access the original model, subjecting it to analysis, re-design, and successive refinement. Stylists can obtain photorealistic visual output directly off the model, toolpaths can be created automatically which will drive machine tools, and draftsman can produce drawings by working directly with the model.

"Drawings are a by-product of the process," he would have said, his nose wrinkling slightly. "Not an end product by itself.

"Who would ever want a drawing as the only output of a design process?" the CAD/CAM veteran would continue, his voice undoubtedly rising in pitch and volume. I mean, who else could use it? How would you get the data from one application to another without having to re-input it?" At this, he would probably shiver as if he had glimpsed a world he didn't want to be part of.

It was crystal clear, at least in the minds of these CAD/CAM veterans, that any user worth his salt would

choose the turnkey system for his work, and only those who couldn't afford the high ticket prices would be forced to settle for the poor imitations of CAD systems offered by the PC software vendors.

For "poor imitations" is exactly what the turnkey vendors believed them to be. These sophisticated, dedicated, and right-thinking representatives of high-end CAD/CAM, collectively looked at the desktop CAD systems and could see no justification for their use, as hard as they tried to see it. Because of this, they completely ignored that piece of the market and allowed upstart companies such as Autodesk to not only grow, but in their success to grab large numbers of their customers, and to never give them back.

Consider the standard, instinctive and universal reaction by members of the turnkey CAD/CAM community to the success of Autodesk. The party line, which was actually deeply and honestly believed, was that AutoCAD was being purchased simply because it was cheap, and that after companies bought it, they put it away and never used it again. This was a popular theme at all of these companies, at every level from consultant to programmer, from salesman to Chairman of the Board. By this reasoning, the failure of Autodesk was only a matter of time as users eventually, but inevitably, came to their senses and returned to the family.

But let's briefly look at the real history of desktop CAD. Consider what a customer got for his money when he bought a turnkey system in the early 1980's. For $100,000 a seat, this customer could design, analyze, refine, view, dimension, and manufacture his parts. The goal has always been to achieve all of this from one single data representation, but this was never achieved. The process wasn't perfect, or even close to perfect as all users knew, but it essentially worked. Compared to old-fashioned, non-interactive or manual methods, it represented a true revolution for that user.

Consider the user of the desktop system of the early 1980's. For $10,000 a seat, the customer could create very complicated engineering drawings, and easily output them to paper, but that was all. Did the desktop system attempt to solve all of the customer's problems, or to provide a total solution? Not hardly! Did you get more for your money from a turnkey system than from a desktop system? Clearly!

Why then would any user be satisfied with a desktop system? How could any company settle for less than a full-solution approach? Shouldn't this be the goal of all users?

Real Users buy Desktop CAD

Well, there were a few small problems with this view of CAD/CAM users.

First, while this top-down, highly elegant and abstract way of looking at design and manufacturing is pleasing to the intellect, and certainly reflected the majority view of upper management of the Fortune 500, it unfortunately had virtually nothing to do with CAD/CAM as it was practiced by the majority of users in 1982.

Keep in mind that, even today, out of approximately 2,200,000 machine tools in existence in the United States, less than 10% of them have a computer attached to them. Computer-controlled machine tools need digital input, but all those others having human operators demand analog input. Typically, the source of this input is from light waves bouncing off a multi-view drawing into a machine-tool operator's eyeballs, and then passing directly into his brain. More often than not, this drawing will have been produced on a desktop CAD system.

Secondly, most parts in this world are designed and manufactured by small shops. You would be amazed to know how little of a General Motors automobile is actually produced directly by General Motors; most is produced under contract to small businesses.

As soon as professional drafting capability became available on PC's, the reaction of users was immediate; they went out and bought it in droves. As a result, the market began to be differentiated, and users with different requirements began to express themselves by buying different types of systems.

At the "low" end, where small, production-oriented companies tend to exist, desktop CAD was an instant success. Generally, these small businesses produce their products according to their own techniques and can't afford, don't want and don't need some gigantic, complicated, totally integrated, corporate-wide unambiguous model representation, when all they need is to produce a drawing, thank you very much.

At the "high" end, large companies bought large systems top-heavy with features. Where the practice of design and manufacturing allowed it, integrated techniques began to be utilized, and designers, draftsmen, and application engineers worked on the same system, all sharing the same database. For them, the turnkey approach was ideal.

Where integrated environments were not available within these large corporations "pockets" of traditional CAD activity continued to exist, and still exist today. These pockets existed in spite of corporate management, not because of them. It was here that desktop CAD found a home.

Within these large corporations, many users attempted to justify the purchase of the expensive turnkey systems on the basis of the enormous return the company would see in automating the production drafting process, but they rarely got a payback on the drafting alone. Before the advent of desktop CAD, many customers in fact found it cheaper, and just as efficient, to go back to pen and paper. With the availability of low-cost desktop CAD, thousands and thousands of users not only found that this was an attractive alternative to the turnkey vendors, but that it was exactly the right solution for their needs. They simply didn't require anything more.

The fact that the turnkey vendors should have seen this years ago is not only testament to their lack of vision but to their obvious ultimate, and totally inevitable fate. The fact that the desktop CAD vendors did see this (looking back, I'm not sure if this was through incredible foresight or simply a keen observation of what was happening, and then a reaction to the opportunity), is evidence that giving the customer what he really needs, and not just what you want to sell him, still counts for something.

In the face of overwhelming evidence, what is it about the turnkey vendors which cause them to believe in their own approach so strongly?

Keep in mind that the turnkey vendors saw themselves having a guiding purpose in life; a mission, really, which was to bring perfection to the design and manufacturing process. The achievement of one single, unambiguous and complete digital representation of real objects, so that a complete simulation of the manufacturing process could be performed, was thought to be within their grasp. The pursuit of anything less than this level of perfection was simply not worthy of their efforts.

Fortunately for us, Autodesk didn't labor under such a burden; making money, and providing useful solutions to satisfied customers seemed to be sufficient motivation.

Drafting vs the Total Solution

Although it may no longer be true in two years, it is relatively safe to say that at this point in the evolution of commercial CAD, one can make a simple distinction between the desktop and the turnkey systems. Desktop systems are drafting systems, generally purchased by users whose needs are local, well-defined and practical,

and whose requirement of that system is principally as an aid in the creation of drawings or other visual artifacts such as renderings. The turnkey systems, on the other hand, are intended by their creators to build complete digital representations of real-world objects. This is called a model.

It is important to understand that, even if levels of technology were even closer than they now are, this would continue to be a fundamental difference between the two. Take away the integrated applications of a turnkey system, and you would be left with a system whose purpose, at least from the point of view of those who produced it, was to provide high-level modeling capability, that functionality having been developed to address the most difficult problems of manufacturing companies, not merely to provide a basic design solution; and certainly not as a system for simply putting lines on paper.

Since the turnkey vendors believe themselves to be in the business of providing modeling capability to be accessed by the world of applications, they universally tend to regard drafting and drawing production as a necessary, but technically unchallenging, therefore unimportant, adjunct capability. I mean, anybody can create a drafting system, can't they? Where's the challenge in that?

The challenge was not technical, of course, but financial, and Autodesk saw this early. Its customers, who are much different from turnkey customers, made it successful and are continuing to do so.

The Autodesk Customer

Based upon our collective experience, we believe that we understand a great deal about the typical Autodesk customer. He or she is probably not an employee of a large, manufacturing-oriented company. Rather, they are more likely to be a member of a small shop, or the manager of a small independent business where the expenditure of even $3,000 is significant.

When you pay for a product yourself, and see the checkbook balance drop, you tend to have a different perspective than if someone bought the system for you. It is more personal, and the commitment to that product, and the company that produced it, is greater.

The Turnkey Customer

Large companies typically spend many millions of dollars each year simply to determine general strategic directions for their companies in CAD/CAM/CAE. These companies typically identify individuals, or groups of individuals, who are vested with the great responsibility of charting that future course within the company. Representatives from Engineering, Manufacturing, Design and occasionally MIS regularly meet in long, intensive sessions about what will be the technological basis for the next generation of CAD/CAM.

What they are attempting to find, of course, is one single, grand solution for the company's CAD/CAM needs. The fact that this process has been going on essentially without change for the last twenty years in the American manufacturing industry is evidence of how important this is held to be.

The CAD/CAM selection committee is a creation of such thinking, and is responsible for the support of a generation of CAD/CAM consultants who continue to recommend variations on the traditional theme to these corporate users. The actual product which is the result of the consultant's analysis or selection committee's

conclusion varies with the consultant's background, the target company's needs, and the weather. But it is ultimately tied to a checklist of features which CAD/CAM selection committees assemble as the ultimate measure of a system's worth.

This feature list is common to those of you who have had some association, however brief, with a turnkey company. The presence of one of these features automatically activates a check mark next to that item on the list, and a tally of checkmarks is made at the end of the process. The majority of check marks does not automatically guarantee the selection of a particular turnkey system, however, since personal bias still plays a great part in the selection process.

See how the selection/purchase process for the turnkey system differs from the same decision for a user of desktop CAD. The purchase of an expensive turnkey system is a decision in which a large number of people have an influence. Generally, unless it is an addition to an existing pool of product, it must also pass the test of the CAD/CAM selection committee, be signed off by the head of the department from whose budget it will ultimately come, and very often by the President of that company.

In addition to being a major purchase due to its large price, the decision is of major import because it is part of the process of determining the strategic direction of that company for its entire future. People who buy drafting systems, even at the corporate level, do so based only on the utility of that system, not having to worry about justifying its strategic impact on the company's future.

This is one of the most important lessons to learn about the difference between turnkey CAD and desktop CAD. Desktop vendors are merely providing useful solutions, whereas turnkey vendors are providing future direction; they are dealing with strategic issues, not just practical ones.

One may disregard the importance of the criteria by which large companies buy CAD/CAM products only at one's own risk. This is actually how large companies spend their money. Drafting systems represent old and developed technology, and fail to cause any excitement in those who are charged with determining future strategy. Future directions must involve advanced technology, and the more advanced, the better. No one discusses drafting systems in the corporate boardrooms of Fortune 500 companies.

Integration as the Key

From one point of view, the goal of the turnkey CAD/CAM vendors (to totally automate, and totally integrate) is certainly worthy of our respect. Taken as a whole, the complete design/manufacturing process, which begins with a concept and ends with a manufactured part, is very complex. In any large company it requires literally thousands of separate steps from beginning to end, including the cooperation of hundred of individuals and tens of different departments.

The gain of even a small improvement in efficiency in any step of the process is absolutely guaranteed to result in a directly measurable increase in the profitability of that company, allowing it to better compete in its marketplace. This, plus the guarantee of an increase in quality of the finished product, is the ultimate promise of automation and is a siren's song that cannot be (and should not be) ignored.

The credit for revolutionizing the world's manufacturing industries must go to the turnkey vendors, of course, and not companies like Autodesk. We have made Autodesk one of the world's most successful businesses, and we affect the practical use of CAD on a scale which the turnkey vendors can still only dream about. But the credit for being there first belongs to them.

The widespread use of CAD/CAM in U.S. industry enabled it to remain competitive with nations such as Japan, which constantly threaten it with their ability to offer America's consumers cheap, but high quality finished products. Further progress is still possible, since the really difficult problems remain yet unsolved.

It is to totally automate and integrate the manufacturing process itself which the turnkey vendors have chosen as their goal. In the following paragraphs, we shall see what the scope of this really is, and begin to realize the scale of the problem which the turnkey vendors have set for themselves.

Design/Manufacturing Process

The totality of the design & manufacturing process is defined by implementation, and differs in detail within every manufacturing company. The following is representative of that process as viewed from within the turnkey industry.

Conceptual Design: Also sometimes called "preliminary design" or "functional design," this stage deals not only with aesthetic issues such as styling, but with practical issues such as simulation and industrial design for manufacturability.

Paper and pencil, brush and oils, and sculptor's clay used to be the conceptual designer's tools in the automotive industry. Today, modern CAD/CAM systems provide him more and more powerful tools which free him from the necessity to create physical models.

It is here that companies such as Cognition, Aries, and Parametric Technologies have seen an opportunity to provide design engineers an entirely new way to approach the design engineering process; offering techniques which lie far beyond traditional methods and allow engineers much greater freedom to exercise their creativity.

Photorealistic rendering output is becoming an essential capability for conceptual design; it allows management to view the design as it would be manufactured, and also allows engineers to try different variations of the design without the accompanying investment in cost and time that normal prototyping techniques traditionally require.

Analysis and Refinement: Also loosely termed CAE, or simply "engineering," various high-level capabilities come under this category.

Finite Element Modeling and Analysis is performed as part of the engineering process. This stage of the process, which is intended to subject a preliminary design to real-world constraints and to iterate on that design until its behavior, given the design, is acceptable. Even within the narrow discipline of FEM/FEA, there are many specialist disciplines. These include fatigue analysis, thermal, vibration and magnetic analysis. Plastics, iso-plastics, and composites complicate the analysis. The exercise of finite-element modeling and analysis is one of the more obvious "applications" to which an existing design is subjected, but there are a large number of others.

Interference analysis, structure design, mass properties, adherence to safety and/or corporate standards and imposition of local codes and regulations are often all requirements for a design to be accepted, and that design generally must pass these analyses before it can be considered for manufacturing or construction.

In the design of an automobile, for example, stress analysis is an issue only for key engine or body parts. More

time-consuming is the ergonomic design of windshields, instrument panels, and even seats. A new water pump must not only be efficient, and deliver so much volume of water per minute, but it must also fit comfortably within the numerous other components which comprise an engine.

Design for Manufacture: Also termed "design modeling," this is another step in "reality design." Often, a so-called "finished" design is impractical to manufacture. Setup costs, consistency with existing manufacturing methods, or excessive complexity may preclude the consideration of an otherwise good design, causing that design to be modified.

A large number of applications exist which satisfy this requirement. The lifetime of a stamping tool, for instance, can have a significant effect on the long-term profitability of a division which manufactures press parts: this requirement alone may have an overwhelming influence on its design. In the plastic injection process, many designs are instantly made infeasible due to their inability to lend themselves to the realistic flow properties of the liquid plastic that is injected into them at high temperatures and pressures. A difference in 5% in injection and cooling time for a complex mold can make the difference between profitability and loss to an industry which works with little room to spare.

Pedestrian considerations such as the design of clamps to hold parts while they are machined, and machine-to-fit tolerances given the practical availability of real machine tools are make-or-break decisions for a manager to make.

Included within this area are assembly verification, component design, and electro/mechanical design.

Drafting and Documentation: This is the world of AutoCAD, yet this area represents but a small part of the turnkey vendor's CAD/CAM universe.

Detail drafting represents no more than one-third of the requirement here. Technical illustration, schematics, and layout are equally important.

Before the days of geometrical models, detail drafting used to represent the "meat" of practical design. Due to the significant limitations of current turnkey design systems, much of detail drafting may never appear on a geometric model.

For example, fillets and chamfers may appear only as "features" on models and may never be represented as actual geometric constructs. As a practical issue, it is far easier to represent a fillet by a symbol on a drawing, and then to cut it with a single path of a ball-end mill, than to go through the difficult mathematics required to represent it geometrically. This is something which practical designers know and make use of.

Other aspects of the detail drafting process have to do with what we regard as "drawing creation," and are intended to aid the ultimate downstream machining process. Surface finish characteristics, tolerance limits, detail magnification, and other aspects of detail drafting are not part of the geometrical model, yet become part of the total representation of the design by virtue of the fact that draftsmen, at least within the turnkey system, can access the original model and work directly upon a local representation of it, even though they are not allowed to modify it. Thus, draftsmen can be specialists in drafting and drawing creation, without having to be expert designers too.

Toolpath Creation & Machining: Also termed "manufacturing engineering," this phase of the process is one of the most complex and demanding. Composed equally of "manufacturing preparation" and "manufacturing simulation," most companies spend the bulk of their CAD/CAM budget here.

Manufacturing preparation includes pattern nesting, tool design, fixture design, sheet metal development, manufacturing quality control analysis, and the actual NC programming itself.

Manufacturing simulation includes coordinate measuring machines, NC flame cutting, off-line robotics, NC tube bending, wire EDM, milling, drilling, routing, flame cutting, turning, and the important area of NC toolpath verification.

Although machining is essentially performed directly off the model geometry, it is by no means as "automatic" as the descriptions of it tend to imply. N/C is still more art than science, and even old-fashioned techniques of creating machined parts have not disappeared.

Creation of geometry is often the simplest aspect of the N/C process. Due to limitations in the algorithms which the turnkey vendors provide, "work-arounds" always have to be provided, including the ability of the user to directly edit the tool path which is being generated.

Toolpath simulation is intended to allow the user to see the form of the finished part that will come out of the machining process, and to correct any problems which are observed. The development and maintenance of postprocessors, which translate geometric toolpath descriptions into a language which each machine tool understands, is an industry in itself.

The Total Solution

Modern turnkey vendors offer solutions for many of these areas, and generally more. Automatic nesting, sheet metal bending, plastic injection mold design, and mold flow analysis, shoe and dress design, and other applications are available within the turnkey vendors bag of tricks. It is their goal to provide the customer's total solution, and it is this which the turnkey vendors have been attempting, over the previous twenty years, to solve.

The Reality

Have the turnkey vendors actually succeeded in the pursuit of this dream? Most users of turnkey systems would say that they haven't. The reasons for their ultimate failure have a lot to do with their history, and their strategy of operation.

Their systems are very large (huge, gigantic, and other descriptors don't really do them justice). Large systems, even if the original design was modern, open, modular, and easy to enhance conceptually, ultimately turn into monsters which are intractable, closed, and which reflect no individual's point of view.

Secondly, the turnkey systems offer just a few solutions to customers' problems, and these solutions are a product of internal development. This not only guarantees that these solutions must be "generic" in the sense that they must attempt to solve everyone's problems, but that they will similarly fail to please all who use them.

Finally, the philosophical approach to the engineering design process reflects thinking which was current when the system was originally created which, in most cases, was a long time ago. The sheer size of these systems makes it a practical impossibility to make significant changes to them. This is why smaller, more focused companies such as PTC are having "sudden" success and are beginning to gain the attention of the users of traditional CAD/CAM/CAE.

Far from providing a totally integrated and automated solution to the process of design and manufacturing, turnkey systems virtually require that each step in the process is separate from the other. This is what the new technologies are intended to address.

The Market

Traditional CAD/CAM Market Segmentation

Segmentation by Revenues, Process, and Price Points

Revenues	Part of the Process	Vendors	Price Points	Market Segment
10%	10%	MSC Analysis, Swanson Analysis. PDA Engineering, SDRC	$50,000 – $200,000	Analysis
2%	15%	PTC, Premise, ICAD, Iconix, Wisdom Systems	$10,000 – $50,000	Conceptual Design
60%	25%	PTC, SDRC, Prime/CV, IBM, McAuto, Intergraph	$12,000 – $60,000	Detailed Design
20%	20%	ACAD, McAuto, Prime/CV, CADkey, IBM, MicroStation	$3,000 – $40,000	Drafting/ Drawing
8%	30%	McAuto, Prime/CV, IBM, MDSI, ANVIL	$12,000 – $50,000	Manufacturing

Source: Wessels, Arnold & Henderson

Figure 1

Figure 1 is a representation of the traditional CAD/CAM/CAE market. As one can see, it covers almost every aspect of the design/manufacturing process as we defined it previously. By the estimation of Wessels, Arnold & Henderson, an institutional research company, the mechanical CAD/CAM market "numbers over 2 million users worldwide and provides a total software revenue potential of $10-$20 billion at a 50% saturation expectation. We estimate the current penetration of the mechanical CAD audience is still less than 20%. More importantly, early users of CAD/CAM systems, using systems based initially on mainframes and minicomputers and later on desktop engineering workstations, are ready for a new level of design capability."

These words were actually written to accompany an analysis of the potential for Parametric Technology's product, but they are as true for Autodesk as they are for PTC. In that analysis, the research company found that the most receptive customers to PTC's products were experienced CAD users who were frustrated with the

disjointed process as actually provided by the turnkey vendors.

The failure of the turnkey vendors to provide integrated solutions to this process is shown by comparing the revenues obtained from a single part of the process versus the time manufacturing companies actually spend on it.

The conceptual design phase represents a relatively small source of revenue, but consumes more than 15% of the design/manufacturing process. According to a British Aerospace study, over 60% of the total money spent on a project is actually committed at the conceptual design stage, when very little is know about what the project will actually entail.

The traditional vendors have not sufficiently addressed this stage of the process, effectively causing the bulk of conceptual design being done with inadequate tools, including drafting and drawing systems.

As newer, more powerful, and more intuitive products become available, we should expect to see a rapid readjustment of spending to begin to match the commitment of actual resources, and the revenues arising from products which address the conceptual design phase should grow rapidly.

Wessels, Arnold & Henderson has studied a new market which they call the mechanical design automation market (MDA), and estimate that this market will grow well over 50% per year for the next six years, up from about $20 million in 1989 to well over $1.0 billion in 1995. They project that penetration of the entire CAD/CAM software-only market will then still be less than 20%. Figure 2 shows the prediction for growth of this market.

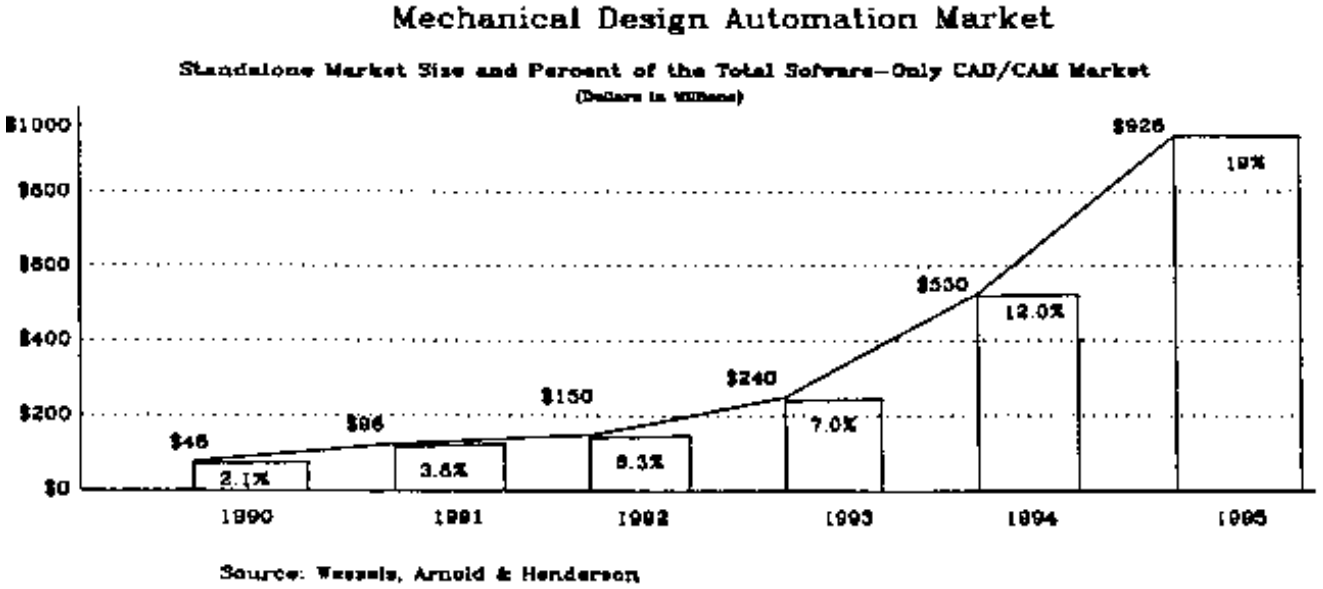


Figure 2

With respect to the drafting segment of the market, they also have a great deal to say, and it is worth quoting directly from their study.

"The Drafting segment of the mechanical CAD/CAM industry is characterized by several new participants, namely Autodesk, who have caused severe dislocation to the large turnkey CAD/CAM vendors who dominated the industry at the beginning of the 1980's. Autodesk, among others, entered the market offering slightly lower functionality for one-tenth the cost ... compared to some drafting programs selling for well over $20,000 per

seat. . . .

"In the future, we estimate the drafting segment to continue to shrink as more of the process moves into the conceptual stage of development, causing further dislocation to the well-established vendors. *Autodesk ought to be able to transition this move with its next generation of products and act as a catalyst for many users who want low-cost replacements for drafting and a strategy for the next several years to conceptual design* (italics mine)."

Mechanical Design Automation

What capabilities must a product have to be part of this projected explosion? The industry itself seems to have already defined the important elements of such a product. According to analysts, consultants, and CEO's of Fortune 100 companies, to even begin to address this requirement a system must include:

- Advanced Modeling
- Parametrics/Constraint Management
- Integration of Iterative Steps/Interactive Analysis

Modeling: The modeling technology required here represents a litany of terms which most of us can recite flawlessly in our sleep. It includes Solids Modeling with NURBS, supporting non-manifold topologies. Feature-based modeling is also becoming an important weapon in this battle of buzz-words.

Parametrics/Constraint Management: Another battle is being fought between those who favor the parametric approach (PTC) or the variational approach (SDRC). Having any sort of capability here, however, is becoming absolutely necessary even to play in the same field as these vendors. Based upon previous buzz-word wars I have seen, we should expect to see an industry-wide mad rush to place this capability, or something resembling it, on the feature list of every CAD/CAM product.

Integration of Iterative Steps: This is a much more difficult goal to achieve than the others because it doesn't depend so much on raw technology, as it does on applications such as FEM/FEA being integrated within the design process. What most systems lack is an interactive method to model, mesh, analyze, and re-model a mechanical part. This capability would provide those who offered it a great advantage in the market.

AutoCAD Designer

At Autodesk we have the ability to build such a product. Much of it could be in place for Release 12, and could be almost complete by Release 13. The philosophy of the Designer product would be that it would offer no options; every piece of advanced technology which we possessed would be part of the system. Additionally, it would contain no drafting or drawing creation capability.

This would allow it not only to leverage off existing AutoCAD installations as a high-end design station which would be 100% compatible with the AutoCAD systems, but AutoCAD could also leverage off the installations

of the Designer product, especially within large corporations who have purchased Designer for "strategic" reasons. At a price of $5000, it could take a big bite out of PTC and Intergraph.

The Designer product must be compatible with AutoCAD, but not just through DXF files, or "translation" routines. To insure this, we must guarantee that the development streams share as much code as possible. In the long term, they should, in fact, be created off the same code base. By utilizing the fruits of the Proteus development, we will be able to provide and support different user interface styles for AutoCAD and Designer with very little extra effort. The databases should be identical as eventually should be the graphics pipelines.

The Designer must be consistent with AutoCAD in offering 3d-party developers with a rich array of tools, including external access to geometrical modeling, constraint management, and rendering capabilities of the system.

The user should be able to execute both AutoCAD and Designer within the same desktop environment, passing data (and geometry) back and forth in a manner transparent to the user. Additionally, selected "accredited" 3d-party applications should also share in this integration, giving the Autodesk customer the ability to create his own custom turnkey environment.

User interface tools, data access mechanisms, and style guidelines will be available to these developers, allowing them to create applications which are truly integrated with AutoCAD.

As an initial marketing strategy, we should probably identify a few key developers whose products would be "integrated" into the AutoCAD Designer, available as options from selected dealers.

Technical Content

The approach to modeling of AutoCAD Designer would allow several simultaneous points of view to exist. The highest level would be of a consistent solids modeler. This solids modeler would be capable of allowing the user to work with solids, surfaces, and/or wireframe without having to drop the solids modeling context.

In addition to the "exact" solids modeler, we would offer the user a conceptual modeling tool, one to be used for "what if" exercises. Offering a facetted representation, this modeler will offer almost instantaneous booleans, and virtually real-time hidden-line removal. This would be especially useful for the conceptual modeling stage where exact designs are not the primary focus. By tracking the design, the system will allow an automatic recreation of the model within ACIS, and allow the designer to polish and refine the model for further modeling/design/analysis work. This facetted modeler is now available and owned by Autodesk.

A 2D Region Modeler ("2D Solids") would be available, allowing the user to define closed planar areas by solids modeling operations (join, difference, etc.). Full support for all wireframe geometry as bounding entities would be supported, including circles, ellipses, and splines. This will provide a more complete representation of objects in 2-D, permitting automated computation of geometrical characteristic such as area, perimeter, centroids and moments.

A translator which would automatically create complete solids models from multiple-view 2D drawings will be available. This would form an important link between the almost uncountable number of existing 2D drawings, and the world of high-end design. This product exists, but would have to be acquired.

A complete surface modeling system would be available, independently of the solids modeling system for

those who were more "traditional" in their approach. Tools which allowed 3D-party access to this modeler could support modern machining applications which would compete favorably with those offered by the turnkey vendors.

An automatic mesher for closed 2D regions would be part of the modeling environment. This would allow for refinement of the mesh, application of loads, definition of constraints, and display of loaded results. In addition to an automatic data interface to popular high-end analysis packages, the system would provide a fast analyzer so that most of the analysis could be done without leaving AutoCAD Designer, thus completely closing the analysis loop. In addition, it will provide a sophisticated basis for 3d-party developers to build more advanced FEM support. The beginnings of this product now exist, although significant development would be required to turn this into a finished product.

A complete 2D variational design interface would be provided, allowing the user to specify designs by variable dimensions and/or constraints. This would allow the user great freedom in his/her ability to create families of parts, to quickly vary designs and see the results, and to create and animate designs for articulated mechanisms. Work on this product is proceeding.

A "poor-man's" parametric design system would be offered which allows the user to define a part parametrically by specifying dimensions symbolically, and attaching values to these dimension at insertion time. The process, which exists and is impressively simple yet surprisingly powerful, automatically generates an AutoLisp program which defines the geometry in terms of these variables. The system can prompt for the values of the variables, or accept default values from a file, creating a block for insertion into the drawing. This capability, with the name "GLISP" (for Generated AutoLisp), is soon to be marketed by our subsidiary in the United Kingdom.

The graphics pipeline of the AutoCAD Designer would support the widest range of display devices, including those capable of maintaining a complete copy of the 3D model internally. This would allow it to take direct advantage dynamic graphics, and internal rendering engines which are becoming increasingly more powerful, and increasingly less expensive.

While the lure of technology is sometimes difficult to resist, resist it we must. AutoCAD became successful through a dedication to providing useful and high-quality products, not by solving the industry's most difficult problems. In the same way, AutoCAD Designer must not be positioned to contain 100% of the capability of competing products. To attempt to do so would put us on the road of diminishing returns. The most technically "challenging" problems require far, far more resources to solve than the revenue one gains in return for it.

By being consistent with the AutoCAD philosophy, which has helped grow one of the industry's most successful companies, we will similarly be in a position to capture a majority of users in the Mechanical Design Automation market. Again, offering well-written, useful products, and providing capability which the user appreciates, succeeds far better than an avalanche of pure technology.

AutoCAD

The future of AutoCAD will not be neglected. In addition to the large number of tasks to be undertaken to help maintain AutoCAD's predominant position in the world of drafting and drawing, AutoCAD would also benefit from the addition of a selected few of these technologies.

The 2D constraint manager, properly integrated, would seem to be an ideal tool for the 2D world, providing additional capability to achieve the goals of our users. The 2D-to-Solids translator also appears to be a natural for

AutoCAD. Even without the ability to further manipulate this model, users would be able to easily and quickly see a fully rendered 3D representation of their 2D drawings, obtain cross-section data, generate sophisticated technical illustrations, and obtain mass properties. All of this would be useful, even if a user never explicitly modeled in 3D. Optional features for AutoCAD could include the NURBS-based surface modeler, the facetted conceptual solids modeler, and the ACIS modeler.

Conclusions

Enhancing the technological content of Autodesk's products is inevitable. Not only do our investors expect it, but our long-term financial health almost requires it. A company which remains stagnant, or which limits its own potential soon loses the confidence of the investment community, as well as those who make strategic decisions for their companies.

By adding this technology into AutoCAD itself we run the very real risk of alienating a large portion of our current users, and encouraging them to migrate toward CAD products which offer capability more precisely suiting their needs. If we were to abandon these users, who have been the foundation of our success, we would have no guarantee that our future success, being based on a user base which we do not understand well.

By differentiating our products, we can achieve the best of both worlds, providing additional leverage for AutoCAD in those environments whose purchases are motivated primarily by design considerations. In addition, the wide success of AutoCAD itself will provide the entry of the AutoCAD Designer into companies who are already familiar with Autodesk, introducing them to high end design, or displacing turnkey systems which are already there.

This strategy is an extension of the AutoCAD strategy. Autodesk has never attempted to solve all of the problems associated with design and manufacturing, in contrast to the turnkey approach. AutoCAD was a product which was narrowly focused to solve one of the stages of the process better and less expensively than the turnkey vendors could do it.

AutoCAD Designer should similarly be narrowly focused within the design stage of the process and not be afflicted by the "total solution" disease. By cooperating with our third-party developers, we will be able to cooperatively offer solutions which address the full range of the process, at a price irresistible to the user community.

Our primary strategic approach must include a firm, unyielding commitment to our current customer base; we must be totally unambiguous about our support of the technology in AutoCAD, and continue to be paranoid about satisfying those users' needs. We should also, however, take advantage of what I believe to be a rare opportunity to deliver a fatal blow to the turnkey vendors (and perhaps to PTC as well) by being a key player in the emergence of a new market for high-end design automation, where we are uniquely qualified to play.

Final Words

The concepts in this paper have been presented to a variety of people, although in much less detail than they were presented here. This includes the Managing Directors of Autodesk subsidiaries and their marketing Directors, Autodesk Product Management, Autodesk Strategic Marketing and, with relatively uneven consistency and little

discussion, to members of Autodesk management.

I would very much appreciate your comments.

Nanotechnology in Manufacturing

Ever since I read K. Eric Drexler's Engines of Creation[317] *I'd been interested in nanotechnology and followed the relevant literature while trying to figure out, as is my wont, how to make a buck out of it. As one of the designers of Xanadu and an ardent supporter of the project, Eric became acquainted with Autodesk when we invested in Xanadu in 1988. When Eric saw a demo of HyperChem at a conference, he immediately realised that the fit between Autodesk and Hypercube, Inc., developers of HyperChem, was potentially very good. When Eric and Chris Peterson spoke to me about this opportunity, I finally saw not only an opportunity for Autodesk to establish itself in the scientific modeling market, but also a way to position ourselves to benefit from the advent of nanotechnology, if and when it emerged, for surely one could not design atomically-precise structures without a molecular CAD system, which is precisely what a molecular modeling package is.*

In early 1989, I prepared this talk about the consequences of nanotechnology to help tilt the balance in favour of the HyperChem deal. The talk was delivered at the Autodesk technology forum on May 10, 1990, before an audience which included Eric Drexler, Chris Peterson, and Neil Ostlund, founder and president of HyperCube, the designer of HyperChem.

This talk has been reprinted by the Foresight Institute,[318] *in* Micro-Times,[319] *and* Mondo 2000.[320]

What Next?
The Coming Revolution In Manufacturing

Autodesk Technology Forum Presentation
by John Walker
May 10th, 1990

These are indeed extraordinary times we're living through.

[317] Drexler, K. Eric. Engines of Creation. New York: Anchor Press/Doubleday, 1987. ISBN 0-385-199973-2.
[318] Foresight Institute Briefing #3
[319] Micro-Times, Issue #101, October 26, 1992.
[320] Mondo 2000, Issue #9, Winter 1993.

Few people are lucky enough to live at a time when their chosen field of interest becomes the center of a technological revolution that changes the face of the world.

We've all shared that good fortune; the success of Autodesk in so short a time is evidence of of how rapidly and how completely the evolution of computers has changed the way design and engineering are done and has reshaped the very terrain on which businesses compete.

Yet all of these remarkable events were predictable, at least in general outline, more than 25 years ago. I remember, like yesterday, the afternoon in the spring of 1968 when, after I first learned about the technologies used in the crude integrated circuits of the time, it hit me. There was no fundamental physical reason you couldn't put a whole computer on a single chip. It was just a matter of engineering, money, and time.

Now I certainly didn't know when it would happen, and I doubt anybody anticipated price and performance reaching their current levels, but the direction and the goal was clear. It made engineering and economic sense; there was no reason it wouldn't work; and each intermediate step along the way would clearly pay for itself in commercially successful products. That pretty much made it inevitable.

This is the kind of reasoning I'm going to be using in this presentation—projecting readily predictable trends forward and asking the question "How far can we go, and what happens when we get there?"

Some of the conclusions that seem almost inevitable have profound consequences that are not just interesting, but important for companies like Autodesk who wish to grow and prosper in the coming years.

What's Been Happening

Let's start by looking at the key trend that's driven the entire silicon revolution. It's really very simple: making things smaller, or in engineer-speak, device scaling.

In the early days of transistors, it was observed that when you made a device smaller, you got the best of everything. It ran faster, it used less power and ran cooler, and since you could pack a lot more of them on a single wafer and the cost of processing a wafer was the same regardless of what was on it, the price per device went down.

After the first primitive integrated circuits were made in the late 1950s, the incentive to miniaturise became even more compelling. By reducing the size of devices further, more complex circuitry could be packed onto each chip. And remember, all chips cost roughly the same to make, regardless of what they do or how complicated they are.

Integrated circuits started out as devices with just two transistors on a die, and progressed to building blocks for larger systems. Finally, in the mid 1970s, practical microprocessors, entire computers on a single chip, appeared. These initial devices were crude, which led many observers to dismiss them as toys. Indeed, none of the major computer manufacturers of the time played a significant role in the development of what is now the universal way of making computers.

It was the chip makers who pioneered microprocessors and developed them to their current state. Why? Because they were familiar with the inexorable consequences of device scaling, and they knew how far, in time, it would carry the microprocessor.

CPU Performance

3466.00	Commodore 128
1598.00	Macintosh Plus
1582.13	Marinchip TI 9900 2 Mhz
66.36	IBM PC/AT 6Mhz+80287
6.29	Sun 3/260, 25 Mhz 68020
4.00	Sun386i/25 Mhz model 250
3.00	Compaq 386/387 25Mhz
2.96	Sun 4/260, Sparc RISC
2.20	Data General MC88000
0.66	DEC Pmax, MIPS processor
0.60	Intel 860, 33 Mhz
0.40	Dec 3MAX, MIPS 3000
0.31	IBM RS/6000

How far have we come?

I have a little program, called the Autodesk Benchmark, that I run on various computers to get a feel for how fast they'll run engineering software like AutoCAD.

Here's an anthology of results, spanning the history of Autodesk, from 1982 through the present.

In 1982, personal computers were becoming seen as useful tools for serious work, but they were still very, very slow for computationally intensive tasks. A typical low cost PC such as a Commodore took close to an hour to run the program. (Putting this in perspective, to do the same job by hand with a pocket calculator would probably take a whole day.)

More expensive and powerful PCs emerged and slowly reduced this time, making more and more complicated tasks practical. Then, in 1984, then 80286 appeared. The impact of this machine on Autodesk can't be underestimated. Calculations that took half an hour on the PCs that existed when we started the company could be done in a minute on the PC/AT. It's no coincidence that Autodesk's sales took off through the roof right about that time.

But it didn't stop there. Three years later, workstations and PCs based on the next generation of chips had cut this time by another factor of ten—from a minute to about five seconds.

And now the newest, shiniest crop of machines just arriving have handed us another factor of twenty—down to less than a third of a second.

In eight years, we've seen a task that originally took an hour reduced, by the simple consequences of device scaling—making things smaller—to less than a third of a second.

This kind of technological progress is hard to comprehend, even if you've lived through it. If automotive technology had advanced an equivalent degree, your car that went 55 MPH in 1982 would today go 615,000 miles an hour, with the same gas mileage and price.[321] You could drive to the Moon in around 25 minutes, if

[321] Columnist Robert X. Cringely later put it this way. "If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost $100, get a million miles per gallon, and explode once a year, killing everyone inside."

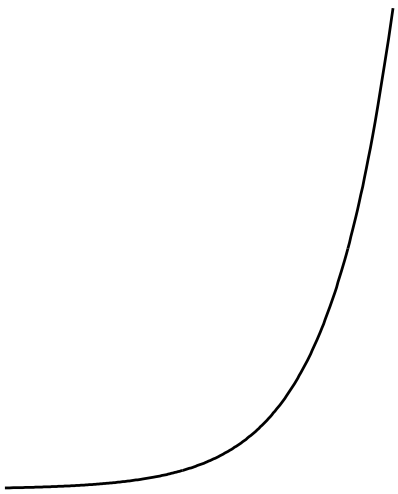
you beat the rush hour and had a good radar detector.

So the inevitable question is. . . just how long can this go on?

There are reasons to believe the end of progress in electronics through pure application of device scaling may be within sight, although not imminent. When the limits of device scaling are encountered, the linear extrapolation that has driven our industry since the 1950s will come to an end and we'll enter another era.

Before getting into that, I'd like to talk about some different perspectives on the kind of exponential growth we've been seeing.

Oh Wow

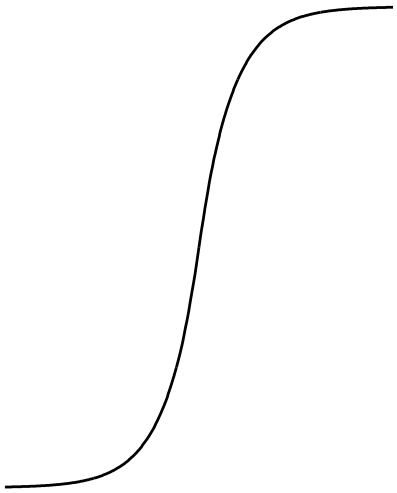


The first is what I call the "Oh Wow" view of things. You look at the curve and draw the obvious conclusion—it's going to go on climbing to the sky forever.

Now I don't mean to disparage this view; most people err on the side of conservatism—they don't realise just how far a trend can go once it's set into motion. But nothing grows forever.

In evaluating any growth trend, whether CPU performance at constant price, world population, or Marin County real estate prices, you have to ask, "what are the fundamental limits to this growth."

Oh Well

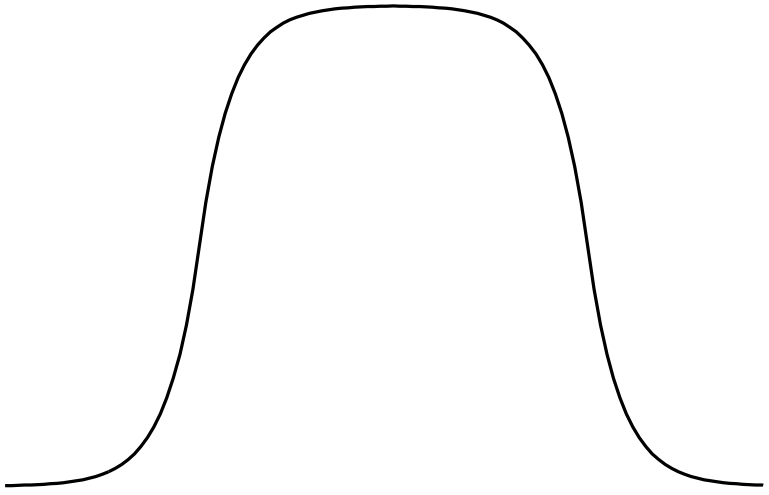


This kind of analysis leads to a different perspective I refer to as the "Oh Well" view. Rather than continued exponential growth, the trend continues until it begins to encounter limits that constrain it, then it tapers off after achieving maturity.

This view is very popular among mainstream business analysts since, especially if you cook up some suitably bogus constraints, you can always justify cutting research and development, reject innovative market expansion and distribution ideas, and relegate the business to a mindless caretaker status once it has reached "maturity".

For example, once you've sold a video tape recorder to every TV station, how many more could you possibly sell?

Oh Shit



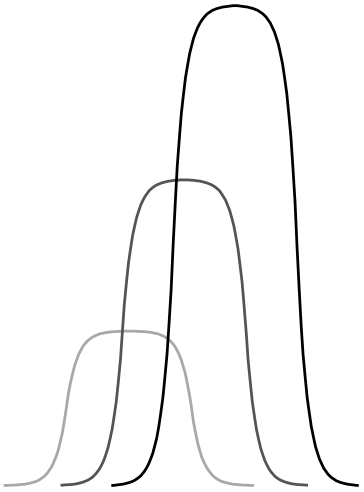
Again, we're not seeing the whole picture. Taking a still longer view gives us what I call the "Oh Shit" perspective.

Nothing in this world is static and nothing, regardless of maturity, lives forever. When a technology, or market,

or species ceases to grow and develop, it's the strongest possible indication that it's become a dead end—its hand is played out—that decline and replacement are only a matter of time.

To understand why, we have to take an even longer perspective.

Oh Yeah!



For this is what's really happening. Progress is made in a series of waves. Each starts with a giddy period of exponential growth. That's when it's fun, when everybody's jumping in, working 24 hours a day, losing money hand over fist, and having a wonderful time.

As the curve really begins to soar, it starts having an impact on the established, mature technologies that went before. This is a general-purpose graph; it's my worldview on a single slide. You can read it as transistors replacing vacuum tubes and then being supplanted by integrated circuits, or of mainframes, minicomputers, and personal computers, of turnkey mainframe CAD companies, workstation based CAD, and mass market desktop CAD software, or for that matter of species diversification and extinction in an ecosystem.

It doesn't matter. It's all evolution in action.

Technological Transitions

Handoff from one technology to the next

Mechanisms	$\longrightarrow$	Electronics
Vacuum Tubes	$\longrightarrow$	Transistors

I call the points where a rapidly developing technology takes off and starts to displace its predecessor "technological transitions." These are perilous times, but they are the times when great industries are founded. Rarely do leaders of the last technology play a significant role in the next; they've usually become encumbered with a bureaucratic superstructure focused on managing a mature market but incapable of acting on the small scale with the rapid pace that's needed to develop its successor—the new market that's inexorably displacing them.

Having grown out of their period of rapid growth, they've forgotten it's possible. They value caution over the very assumption of risk that built their industries in the first place, and through caution, they place at risk everything they have.

Technological transitions are great times to make gobs and gobs of money. Autodesk got on board a relatively minor technological transition in 1982—the second subwave of personal computers, which was a ripple of the microprocessor surge, itself part of the semiconductor tide, contained within the automation industrial revolution. Even so, we managed to turn a hundred thousand dollars into more than a billion in less than eight years.

Just imagine what you could do with a real industrial revolution.

Five Industrial Revolutions

Tools	2,000,000	B.C.
Metallurgy	3600	B.C.
Steam power	1764	
Mass production	1908	
Automation	1946	

Everybody has their own list of industrial revolutions, and here's mine.

A couple of million years ago we figured out that bashing things with rocks you'd worked into special shapes was better that bonking them with any old random rock you picked up. Suddenly you had tools, craftsmanship, carpentry, weapons, wealth beyond imagining, and global proliferation of a previously obscure critter. This was an information revolution: using knowledge to transform existing materials into useful forms.

All of this was based on natural materials, picked up from the Earth or taken from plants and animals. Then, about 5600 years ago, on a sunny Thursday morning, somebody figured out how to extract copper from yucky looking rocks. Now people had access to new materials—technology was no longer limited by what was lying around; it could make new substances and build with them. This led to bronze, iron, alloys, alchemy, chemistry, and steel. This was a material revolution; enabling new technologies by creating substances not found in nature.

All of industry until the 18th century was powered by the energy of human or animal muscles, or natural energy sources like falling water and the wind. This limited both the scope and scale of what could be done. The advent of practical steam power swept away these limitations, spawning trains, steamboats, satanic mills, and capitalism. This was an energy revolution.

As the scale of industry grew, economies of scale could be realised by standardisation and interchangeability of parts. These trends ultimately led to an entire industrial system focused around mass production of largely identical objects. This is harder to date. I use 1908, the date of the first automobile assembly line, as the milestone of mass production. Mass production was essentially an information revolution: it embodied a uniform set of specifications in huge numbers of objects, thereby reducing their cost so many more people could afford them than ever before.

I consider automation to be the most recent industrial revolution. Until the advent of mechanical, electrical, and electronic computers in the twentieth century, any computation or information processing required the attention of a human being and necessarily proceeded at the pace a human could work. The computer revolution, which I

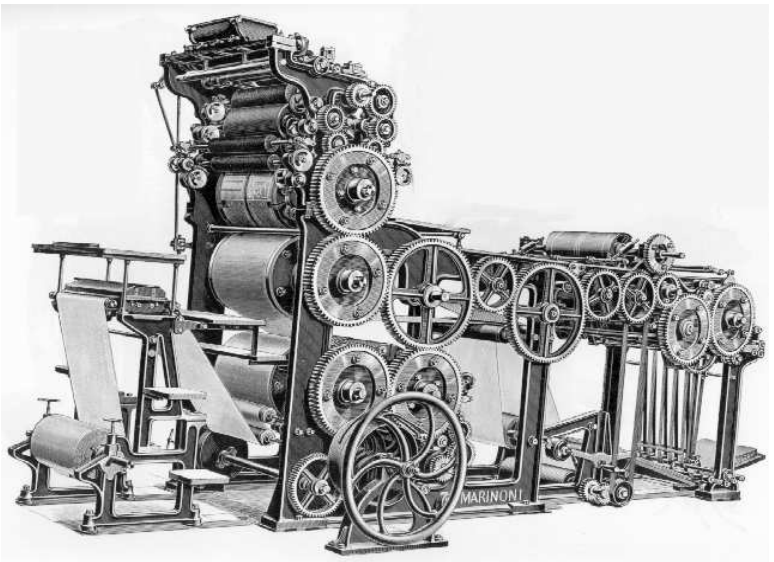
date here from ENIAC in 1946, has been an information revolution that has transformed not only the mechanics of industry, how we make things, but also the structure of our organisations and societies. Ironically, mass production, an essential precursor of automation, is becoming less important as the introduction of intelligence throughout the manufacturing process allows more flexible forms of production.

Metre Scale

Before I get into the details of what's about to happen, I want to make sure we understand the territory. We're about to discuss things that range in size by a factor of a billion to one, so it's useful to go over the distance scale so we don't confuse millions and billions the way the politicians do.

Let's start with the measure of all things, a human being. Humans are on the order of metres in size, actually closer to two metres, but we can ignore ones and twos when talking about factors of 1000, as we're about to do.

Most of the history of technology has been built to this scale, and tended to look something like this.



If an object has to be assembled by people, in many cases powered by people, and operated by people, it doesn't make any sense to make it smaller than people can reasonably use, notwithstanding the design of modern car stereos.

The size of mechanical parts is governed by the materials that compose them and the scale of the machines used to fabricate them. Both imposed severe limitations on miniaturisation throughout most of history, limitations that were surmounted only with great difficulty and expense when absolutely necessary, as in the design of watches.

Wherever technology leads us, effective design at this scale will remain important as long as humans use the products. This is the domain of user interface and ergonomics.

Metres To Millimetres

If we shrink down by a factor of a thousand, we arrive at the millimetre scale, where an ant is pretty big stuff. This is about the limit of what the human eye can effectively see unaided, or the human body can manipulate without mechanical assistance, so it's a convenient milestone on the road to Lilliput.

Here's an example of millimetre scale technology. This, for those of you too young to remember or old enough to have had enough bad experiences and deliberately forgotten, is a vacuum tube.[322]

If you examine this device closely, you'll see that its fundamental geometry: the spacing of its grid wires, the distance from the cathode to the plate, are all on the order of millimetres. With the development of electronics, device scaling immediately became important: the smaller you made a tube, the faster it ran and the less power it used. Unfortunately, the fact that tubes had to be assembled from separate metal parts limited how much you could shrink them.

This particular tube is a 6SN7. That's the type used in the flip-flop circuit of the ENIAC, the world's first electronic digital computer of 1946. Each tube, along with a handful of other parts, stored one bit of computer memory—RAM. The ENIAC contained 18,000 tubes like this, occupied 3000 cubic feet of space, and required 140 kilowatts of electricity. It had about the computing power of a pocket calculator. It was a miracle of millimetre technology.

Tubes like this aren't even made in the United States any more; this one came from the Soviet Union and cost $13. Interestingly, that's almost exactly the current price of an 80 nanosecond 1 megabit dynamic RAM, with a million times the storage capacity and 125 times the speed.

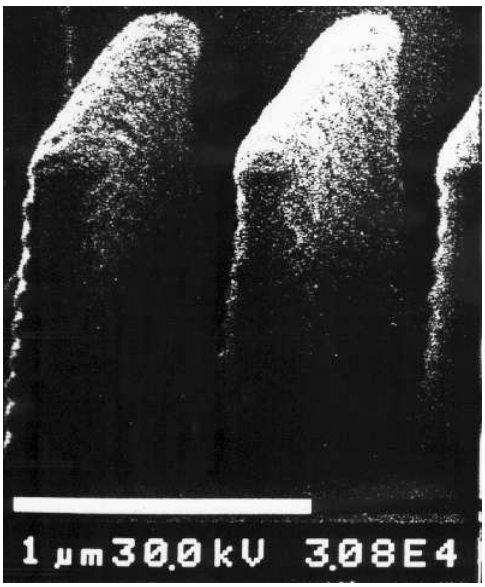
Millimetres To Micrometres

Which brings us to the next factor of 1000 in size, to the micrometre range that's the heart of microchip technology. All of these RAMs and ROMs and microprocessors have feature sizes on the order of micrometres. The entire march of processor technology I showed in the timing slide is essentially the story of learning to shrink our circuits from the order of tens of micrometres to single micrometres, and the frontier of electronics in the next several years will be shrinking further, down to fractions of a micrometre.

And then what?

Well, that's a very interesting question.

[322] I've never been averse to using props during presentations. At this point I whipped out an actual 6SN7 I'd bought a couple of days before at Electronics Plus. Unfortunately, neither nanotechnology nor multimedia have as yet advanced to the point where I can make a vacuum tube pop out of the page as you read this. But, you can turn to page 170 and see a schematic of the circuit I'm talking about. All is one. Or something.



Here's a piece of silicon that's been machined into slabs a third of a micrometre wide. This is about the best we can do right now with conventional processing technology, and to give an indication of just how tiny that is, each of those slabs is smaller than the wavelength of deep violet light.

Needless to say, this image is from an electron microscope, not an optical one.

All the way down to micrometres, we've been able to design circuits essentially the way we did in the days of vacuum tubes. Yes, the devices and fabrication technologies changed, but the rules of the game like Ohm's law remained the same.

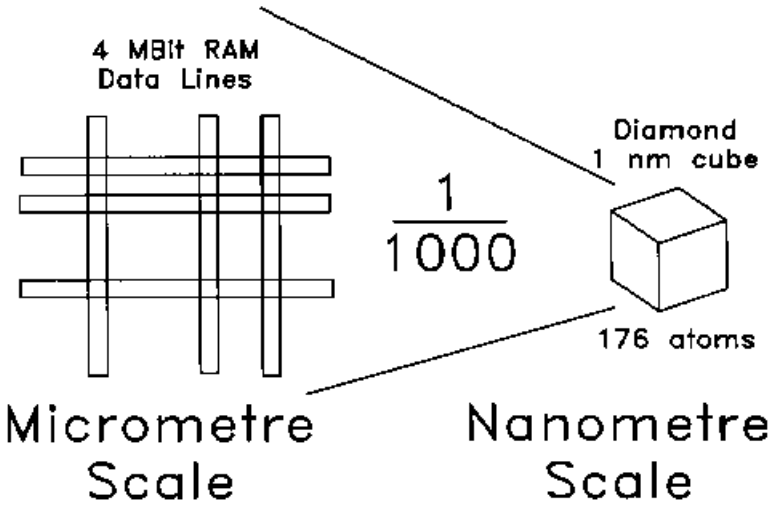
Somewhere between half a micrometre and a tenth of a micrometre, these comfortable assumptions begin to break down. The weird world of quantum mechanics, where the wave nature of the electron becomes apparent, begins to become manifest at this scale and straightforward shrinking doesn't seem likely to work.

This suggests we're already uncomfortably near the top of the curve for conventional electronics.

Micrometres To Nanometres

But how far are we from the theoretical bottom?

Just about another factor of a thousand, it turns out.



If we shrink from a micrometre to a thousandth of that, a nanometre, we've reached the scale where atoms become tangible objects. A one nanometre cube of diamond has 176 atoms in it.

Designing at this scale is working in a world where physics, chemistry, electrical engineering, and mechanical engineering become unified into an integrated field.

This field will be called molecular engineering, and I believe it will be at the heart of the next two industrial revolutions.

Plenty Of Room

> "The principles of physics, as far as I can see, do not speak against the possibility of maneuvering things atom by atom. ...it is interesting that it would be, in principle, possible (I think) for a physicist to synthesize any chemical substance that the chemist writes down. Give the orders, and the physicist synthesizes it. How? Put the atoms where the chemist says, and so you make the substance."
>
> —Richard Feynman, 1959

Over thirty years ago Richard Feynman pointed out that physicists knew no limits to prevent us from doing engineering at the level of atoms. His words are as true today as the day he spoke them.

Until recently, though, while the lack of physical limits was accepted as commonplace, molecular engineering was thought of as impractical, unnecessary, or requiring breakthroughs in knowledge and technique that placed it somewhere in the distant future.

Many visionaries intimately familiar with the development of silicon technology still forecast it would take between 20 and 50 years before molecular engineering became a reality. This is well beyond the planning horizon of most companies.

But recently, everything has begun to change.

Writing With Atoms

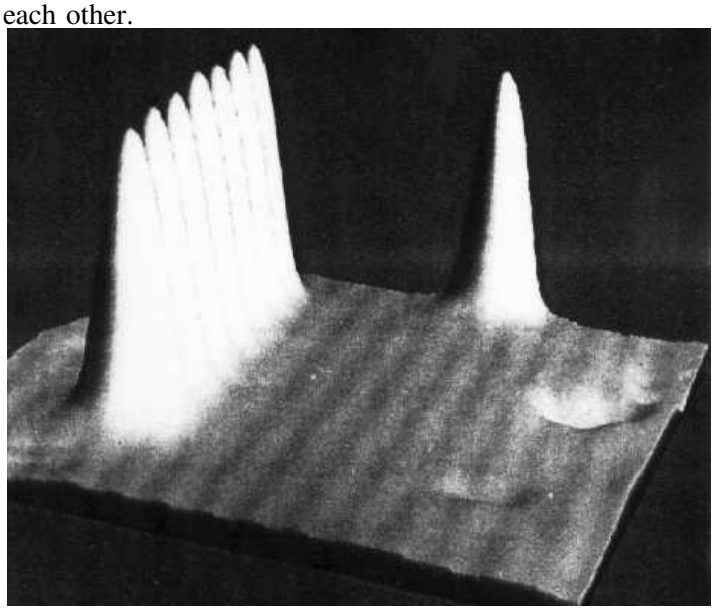
In 1981, Gerd Binnig and Heinrich Rohrer of the IBM Zürich Research Laboratory invented the Scanning Tunneling Microscope. This device, easily one of the most elegant and unanticipated inventions of the century, allowed imaging of individual atoms, and won Binnig and Rohrer the Nobel Prize in Physics for 1986.

In 1985, Binnig and Christoph Gerber of IBM Zurich, along with Calvin Quate of Stanford, invented the atomic force microscope. This allowed imaging nonconductive matter such as living cells to molecular (although not currently atomic) resolution.

Since then, every year has seen new inventions in the rapidly growing field of scanning probe microscopes. They're now imaging bits on magnetic surfaces, measuring temperature at microscopic sites, and monitoring the progress of chemical reactions.

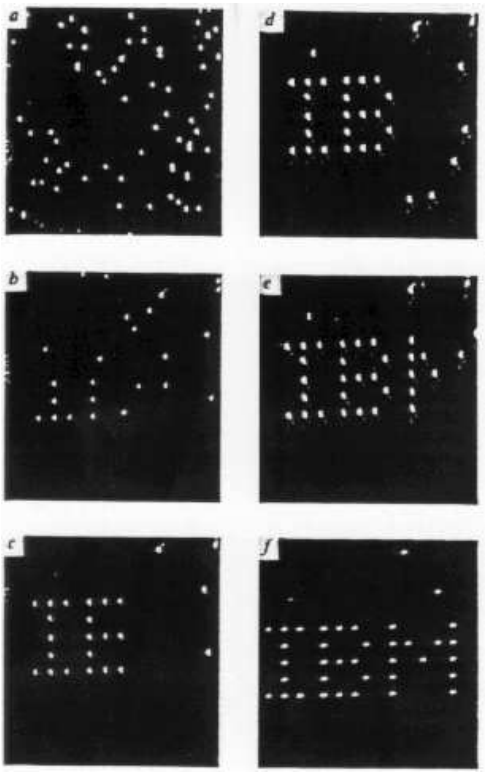
Recently, IBM San Jose used a scanning tunneling microscope to, in Feynman's words, put the atom right where the chemist says.

Here's a picture of xenon atoms on a nickel crystal, lined up in a row by pushing them into place with an STM tip. Remember, those bumps are individual atoms, and they've been moved precisely into position, in a row, one half nanometre from each other.

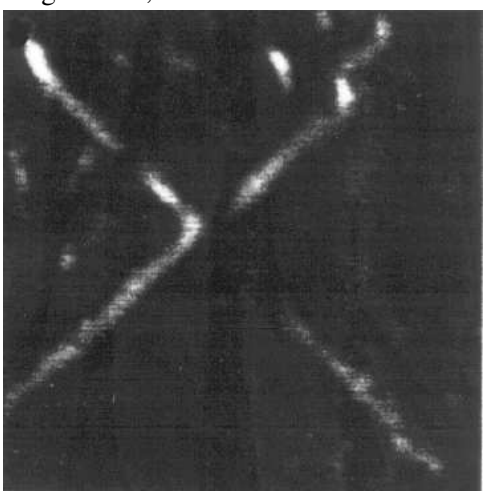


You can not only put atoms right where the chemist wants them, you can put them right where the marketing department says to, as well.

Here, from the same paper, is a sequence of pictures showing a bunch of randomly positioned xenon atoms being aligned to spell out the name of the sponsor.



Again, each dot in this picture is a single atom, and the letters are 5 nanometres tall.



Here's an illustration from a paper published a few weeks before the IBM result. The bottom picture shows an X drawn with an atomic force microscope on material adsorbed onto a zeolite crystal surface. The X is 8 nanometres tall, and it remained intact for the 45 minutes it was monitored after being drawn there.

This experiment was done with a desktop instrument no bigger or more complicated than a compact disc player.

Molecular Engineering

"...thorough control of the structure of matter at the molecular level. It entails the ability to

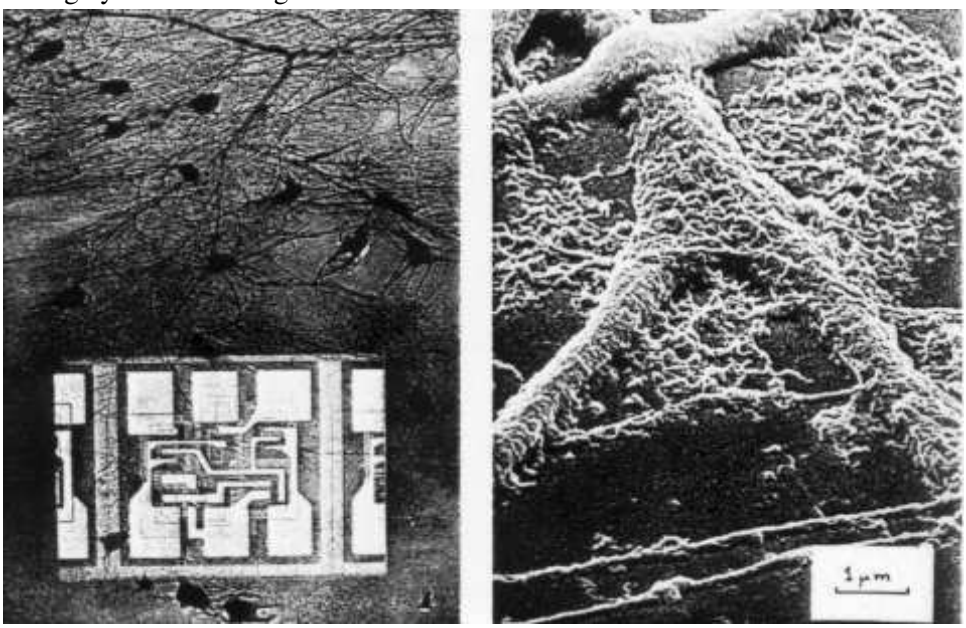
> build molecular systems with atom-by-atom precision, yielding a variety of *nanomachines*. These capabilities are sometimes referred to as *molecular manufacturing*.
>
> —K. Eric Drexler, 1989

> A capability never before available to technology. Previously, the exclusive domain of biological systems.

So what we're talking about is making the next big jump to building systems a thousand times smaller than the ones we're making today; to go all the way to the bottom and start working with individual atoms.

This is called molecular engineering, or nanotechnology. Eric Drexler defines this as control of the structure of matter at the molecular level, however achieved.

Technology has never had this kind of precise control; all of our technologies today are bulk technologies. We take a big chunk of stuff and hack away at it until we're left with the object we want, or we assemble parts from components without regard to structure at the molecular level. Precise atomic level fabrication has previously been done only by living biological organisms. We are entering an era when some of the barriers between engineered and living systems will begin to fall.



In fact, we're already building components on the scale of biological systems. The picture on the left shows a neuron net from a human brain with an integrated circuit component inset at the same scale. The picture on the right is a synapse—the interconnection of the wiring in the human brain, with a one micrometre scale. Remember that this is just about the feature size of the wires in our integrated circuits.

The huge difference in capability between engineered and biological systems is not just the materials from which they're made, it's that the fine structure of the integrated circuit stops with what you can see: there's nothing down below. Since we're forced to fabricate our circuits from bulk material, from the top down, they must be essentially two-dimensional. Biology builds its structures from the bottom up, at the molecular level, and in three dimensions.

Engineers are beginning to learn how to do this.

Many Challenges—But Many Paths To Follow

- Biochemistry: Custom protein design
- Chemistry: Molecular recognition
- Physics: Scanning probe microscopy
- Computing: Molecular modeling
- Engineering: Molecular electronics
- Engineering: Quantum electronic devices
- Engineering: Nanocomposites

The challenges in scaling another factor of 1000 shouldn't be minimised. Developing millimetre scale electronics and micrometre scale integrated circuits wasn't easy, either, but after we overcame the initial obstacles, both progressed much faster and further than anybody initially expected.

The remarkable thing about molecular engineering is that it looks like there are many different ways to get there and, at the moment, rapid progress is being made along every path—all at the same time.

In 1988, a group at duPont led by William deGrado designed a new protein, called α_4 from scratch, and manufactured it in their laboratory. This protein, which never existed in nature, is more stable than natural proteins its size. Researchers around the world are now looking at proteins as molecular structures they can design and build, just as an IC designer lays out a chip.

Chemists are making progress in designing and synthesising molecules that bind to other molecules at specific sites, facilitating the kind of self-assembly that occurs in biology. The 1987 Nobel Prize in chemistry was awarded for just such work.

I've already alluded to the feats accomplished so far with scanning probe microscopes. We now have a tool that lets us see and move individual atoms. STMs have also been used to pin molecules to a substrate and break molecular bonds. John Foster of IBM and Eric Drexler have suggested in a recent paper in Nature that attaching custom molecules to the tip of a scanning microscope may allow assembling objects with up to 10,000 molecular pieces, with atomic precision.

The ability to model and simulate complex molecular systems has been growing rapidly in recent years, driven both by advances in raw computing power, but also by the development of better simulation techniques that now permit modeling of proteins composed of thousands of atoms.

Physicists and electrical engineers are making rapid progress in fabricating electron devices that work at the molecular level. In the past two years, Texas Instruments and Bell Labs have reported molecular-scale quantum transistors and have fabricated quantum wires with X-ray lithography. These quantum wires are on the order of 30 nanometres wide.

Materials scientists and mechanical engineers are fabricating new materials called "nanocomposites," made up of individual particles ranging from 100 to 1000 atoms. These appear to have electrical and mechanical properties unlike any other engineering materials and may prove useful in the near future.

Steps Along the Way: Molecular Electronics

- Speed: 1000 Ghz (100–1000$\times$ present)

- Power: (100–1000× less power)
- Compatible with photonics

Even technologies with enormous potential can lie dormant unless there are significant payoffs along the way to reward those that pioneer them. That's one of the reasons integrated circuits developed so rapidly; each advance found an immediate market willing to apply it and enrich the innovator that created it.

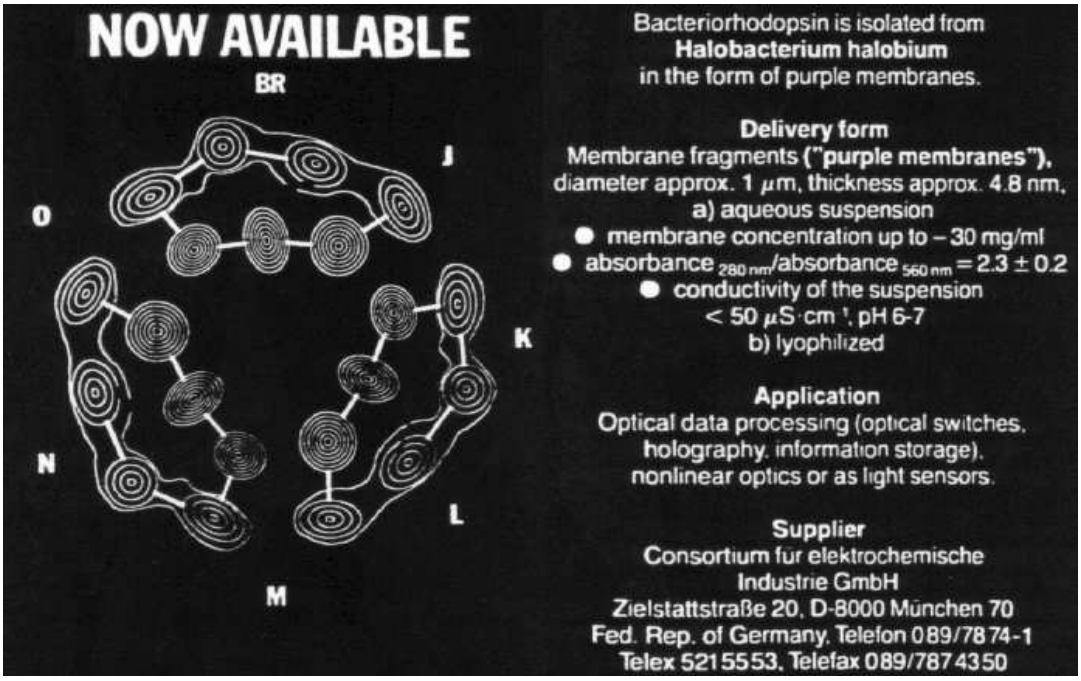
Does molecular engineering have this kind of payoff? I think it does.

Remembering that we may be less than 10 years away from hitting the wall as far as scaling our existing electronics, a great deal of research is presently going on in the area of molecular and quantum electronics. The payoff is easy to calculate; you can build devices 1000 times faster, more energy efficient, and cheaper than those we're currently using—at least 100 times better than exotic materials being considered to replace silicon when it reaches its limits.

Bacteriorhodopsin Optical Memory

- Purple membrane from Halobacterium Halobium
- Bistable red/green switch
- In protein coat at 77K, 10^7–10^8 cycles
- 10,000 molecules/bit
- Switching time, 500 femtoseconds
- Monolayer fabricated by self-assembly
- Speed currently limited by laser addressing

As an example of current work, consider the molecular optical memory research underway by Prof. Robert Birge and his group at Syracuse University. Using the purple membrane from the bacterium Halobacterium Halobium, they've made a working optical bistable switch, fabricated in a monolayer by self-assembly, that reliably stores data with 10,000 molecules per bit. The molecule switches in 500 femtoseconds—that's 1/2000 of a nanosecond, and the actual speed of the memory is currently limited by how fast you can steer a laser beam to the correct spot on the memory.



Lest you think this is some far out distant future research topic, here's an ad from a couple weeks ago by a company in West Germany offering bacteriorhodopsin for sale, listing under applications, "Optical data processing, optical switches, holography, information processing, nonlinear optics, and light sensors."

Sixth Industrial Revolution: Molecular Engineering

- Fine-grain control over the structure of matter
- Materials inaccessible through bulk processing
- 100 times smaller (at least)
- 10,000 times cheaper (at least)
- 1000 times faster (at least)
- Fabrication of quantum devices

And so, we can begin to see the outlines of the sixth industrial revolution: moving from micrometre scale devices to nanometre scale devices. Current progress suggests the revolution may happen within this decade, perhaps starting within five years.

Neither the events nor their consequences will be subtle. Suddenly, we'll acquire new capabilities comparable to those of electronics or computers.

What can we make with it? Well, anything we can design and model that's built of atoms. Think about that. And that includes the essential component of the industrial revolution that will follow, perhaps in the same year.

Seventh Industrial Revolution: Replicating Machines

- Molecular engineering a prerequisite
- *Required* to mass-produce at the atomic scale
- Existence proof

For once we've mastered the essential technology of life, assembling objects at the molecular level with molecular machines, there's no reason we can't rapidly exploit the central trick of life as well: getting the job done with machines that make copies of themselves.

Mass production has reshaped our industries, lives, economies, and societies, but it's been a limited form of mass production: one where the process of production was explicitly designed and rigidly oriented to making a given object.

Once we can build molecular machines, we can design machines that make copies of themselves. By doing so we achieve the second level of mass production: being able to make anything we can design at a cost fundamentally constrained only by the materials and information it contains.

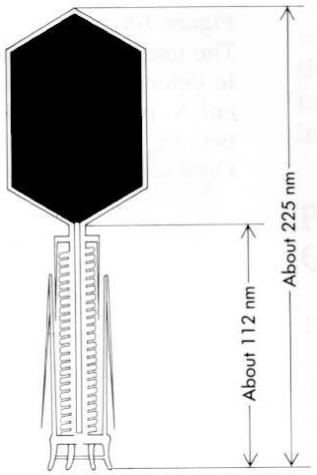
If this seems absurd, just imagine how an engineer at the start of the twentieth century would have reacted to a description of photolithography, the technology we now use to make integrated circuits and printed circuit boards. "You're telling me you can manufacture objects in the millions just by making *photographs* of them? Give me a break!"

In fact, if we want to make objects on the metre scale with molecular engineering, we're going to *have* to design replicating machines. The vacuum tube I showed you has about 10^{23} atoms in it, and if you try to build something that large atom by atom, it's going to take pretty long. If you add an atom every second, it'll take 10^{23} seconds which is a real problem because that's a million times longer than the current age of the universe. But, if you can get your molecular machines to crank out copies of themselves, you can set up a chain reaction that can generate numbers on that scale quite rapidly. That's how biology manufactures bacteria, butterflies, and buffaloes, and it works very well.

This is flexible manufacturing taken to the logical limit. An invention made on Monday could, by the following Friday, be in mass production, with billions of copies fabricated. It's the ultimate triumph of information over machinery, of software over hardware, of intellect over capital.

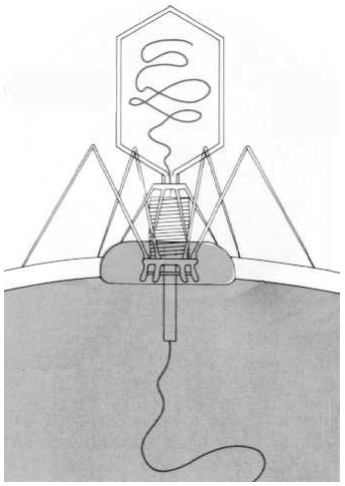
The consequences of this are truly hard to grasp. Just as the development of computers made many problems that have vexed mankind for centuries essentially trivial, we're looking here at the first fundamental change in the means of production in the last two million years. Our economic and societal structures have evolved around assumptions that will no longer be valid once technology reaches this milestone. And it may happen in the next ten years.

But the real question I haven't answered yet is this. "Is it actually possible to make these little tiny machines out of atoms and then get them to replicate themselves, or this all just a pile of hooey, as ridiculous as, say, putting 16 million transistors on a piece of silicon the size of your fingernail?"



Let's look at how we might design such a machine. Here's a little gadget that looks like a lunar lander, but considerably smaller. It stands about 225 nanometres high. This device is designed to operate within a living system, to seek out cells of a particular type, land on them by extending its landing legs, then inject them with material stored in the tank at the top.

You could design something like this, for example, to locate cancerous cells in a human body and kill them.



Here's our little machine attached to a cell, with its injector poked through the cell membrane and emptying the tank into the cell's interior.

For a sense of scale, this entire gadget, standing on its tippy-toes would be less than a quarter the size of the *smallest* feature of a current microprocessor integrated circuit.

Could something like this be built?

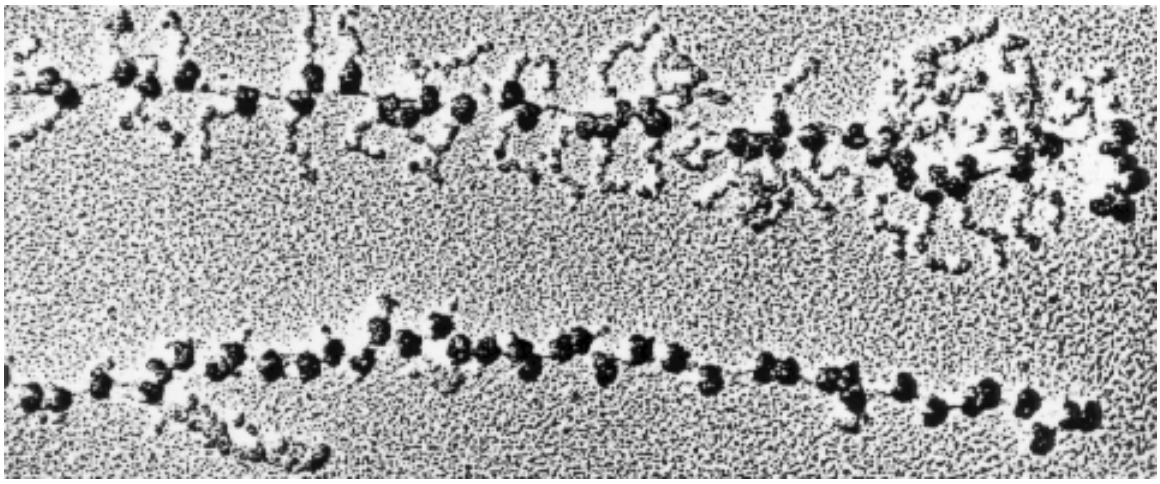
Could it possibly work?

Yes. Here's a scanning electron microscope picture of the actual device. It wasn't designed on a CAD system; it evolved in nature. It's called bacteriophage T4. It's a virus that preys on *E. Coli*, a common bacterium that lives in the human intestine.

So molecular devices exist, work, and even succeed in replicating themselves in a proper environment.

Here's an electron micrograph of some *E. coli* cells infected with another virus, phage f2. The crystalline lattice in the corner of the cell is an array of self-assembled copies of the virus, manufactured within the cell.

You might also ask, "Can we store information at the molecular level?"



The answer to this is also yes, and each of us is living proof.

Here's a picture of a molecular-level file copy operation in progress. This is a 140,000 times blow-up of a strand of messenger RNA being transcribed into the proteins it encodes by a bunch of ribosomes—they're the little beads on the string of RNA. The bumpy strings coming out the sides are the protein chains being assembled by the ribosomes. Each ribosome is a little molecular machine about 20 nanometres across that manufactures proteins to order by reading a tape called messenger RNA and assembling the protein it describes according to the genetic code.

There are quadrillions of them busy at work in each of your bodies at this very moment, and they're about as accurate in copying the molecular data in your DNA as a typical hard disc drive.

Since these biological molecular machines obviously exist, and clearly work, and since biology follows the same laws of physics and chemistry we apply as engineers, there's no question our molecular machines will work. All we have to do is figure out how to design and build them.

And with the rapid progress on all fronts toward molecular fabrication, design may become the most important part of the puzzle.

The Argument For Design

- If you can't model it, you can't build it
- Molecular CAD an essential enabling technology
- Autodesk uniquely positioned
- Technological leverage

The consequences of any industrial revolution extend far beyond the domain of scientists and technologists. Indeed, they are difficult to even imagine. For example, try to envision the present world without any electronic devices. You can't. Nobody can. It would be a very different world from ours.

Similarly, the world that will develop after the next two industrial revolutions is difficult to imagine starting from today, but there is every reason to believe the consequences of molecular engineering will be even more profound in every way that those that followed the development of electronics or computers.

One thing, however, is clear. Unlike all of the industrial revolutions that preceded it, molecular engineering requires, as an essential component, the ability to design, model, and simulate molecular structures using computers.

Computer aided design has been largely an adjunct to engineering so far. It has promised productivity gains, cost savings, and other benefits but except at the leading edge, as in VLSI design, has rarely been an indispensable component of the design process.

But with molecules, if you can't model it, you can't build it. Computer aided molecular design is one of the key enabling technologies of these imminent industrial revolutions, and stands to benefit both by helping to bring them about, and to profit from the fruits of their success.

A computer aided design company that comprehends, throughout the organisation, what is about to happen and takes the small, cautious, prudent steps today to position itself to ride this next wave stands an excellent chance of emerging as one of the dominant industries on the planet as molecular engineering supplants our present technological base to the degree that integrated circuits have replaced vacuum tubes and electricity has displaced steam engines.

"Leverage" is commonly used these days in a financial sense to mean risk/reward amplification through the assumption of debt. But there are many kinds of leverage. Technological leverage, the power born of knowledge, is supreme among all forms. We used it to build Autodesk into the company it is today, fending off competitors with far more money and people by simply knowing where technology had to go and hitching a ride.[323]

Autodesk is, at this moment, the preeminent global force in computer aided design. Around the world, technology is poised at the threshold of access to capabilities scarcely imagined a decade ago, and computer aided design is an essential component of this next chapter in the human adventure. The technological leverage of this next industrial revolution is ours, if we want it.

The lever pivots on the micrometre technology of the microprocessor. At one end, we reach to grasp it with our metre scale human hands, at the other, it manipulates atoms to designs born of the human imagination. With this lever, and the knowledge, courage, and vision to operate it wisely, we can truly move the world.

[323] See the in-depth discussion of technological leverage in "The New Technological Corporation" on page 464.

Max Q

While "The New Technological Corporation" focused on largely on the theory underlying the special strategies appropriate to a company like Autodesk, this memo, circulated to senior management in the fall of 1990, concentrated on details. It seemed obvious to me that the boom of the 1980's could not go on forever, and that there were specific danger signs on the horizon. It's much easier to see trouble on the horizon than to know precisely when it will arrive, but fortunately a company in Autodesk's position need not forgo opportunities in order to protect against turbulence.

To: Central Committee
From: John Walker
Date: September 7th, 1989

Subject: Autodesk at Max Q

> **Max Q.** In aerodynamics, particularly rocketry, maximum dynamic pressure; the moment when velocity, trajectory, altitude, and ambient atmospheric temperature and pressure combine to exert the maximum stress upon the vehicle. Most catastrophic failures occur at this moment.

Viewing the history of Autodesk's growth, as outsiders do, by examining a chart of sales, earnings, units installed, market share, stock price, or any other aggregate metric of performance gives the impression that Autodesk has achieved its success with relatively little difficulty. Indeed, other than blips in the stock price triggered by exogenous events such as the Crash of 1987,[324] almost every measure of Autodesk's performance is a monotonically increasing curve, devoid of both gut-wrenching plunges and giddy, unsustainable spurts of growth. It's the kind of performance that tempts one into believing that change of this magnitude can be managed—that one can guarantee this kind of performance in the future—forgetting that we're simply trying to ride a wave of technological change without drowning, slipping off into stagnant backwaters, or being dashed on the rocks by excessive assumption of risk.

Having lived through the hundreds of thousands of individual events which collectively add up to the numbers on the chart, we know very well that nothing about Autodesk's success came easily. Building the company required ignoring conventional wisdom, willingness to improvise in the face of inadequate resources, and to do whatever was necessary to make our products meet our customers' needs, even if it meant undertaking technical tasks rarely attempted by "application vendors" or scratch-building entire channels of distribution, training, and support where none existed before.

[324] See page 404.

Management by lack of alternatives

And yet, in one sense Autodesk has had it relatively easy since 1982. When people ask me about the "key strategic decisions" made while I was involved in managing the company, my response is that most of the crucial decisions: the ones which, in retrospect, were central in achieving the market position we have today, were not at all difficult to make. In fact, most of those decisions were arrived at simply because they were essentially the *only* courses of action open to us at the time. Other paths leading to alternative destinies for Autodesk were foreclosed due to lack of resources, prerequisites, or imagination. I refer to this as "management by lack of alternatives" and, although not conducive to one's being perceived as a super-manager, it has served Autodesk well.

One of the reasons so many decisions seemed obvious, and that Autodesk prospered from pursuing opportunities in a straightforward manner is that throughout our company's history the fundamental economic and technological trends that were in effect when Autodesk was founded have remained intact. The 1980s were a time of enormous, indeed mindboggling, change. Autodesk was organised at the absolute bottom of the 1982 recession—one of the most severe economic slumps since the Great Depression. As Autodesk's incorporation papers were being prepared and filed, the Dow Jones Industrial Average dipped below 800, marking a low point never reached since. Unemployment was over 10%, interest rates were declining from their 1981 peaks that shattered all records since the Civil War, and gloom was everywhere. Polish workers, near open rebellion against communist rule, were suppressed by a military junta installed to preempt Soviet intervention. For the first time since the 1950s, sober observers in both the East and West began to think of nuclear war as a real possibility. And IBM, to the bafflement of most onlookers, chose a virtually unknown architecture, the Intel 8088, an obscure operating system, MS-DOS, and brought to market a personal computer for which there was no application software.

Our perception that the advent of the mass-marketed personal and office computer, whatever the source, would create an enormous demand for application software led us to create Autodesk in 1982. In a way, even the founding of the company reflected a lack of alternatives. My prior company, Marinchip, was expiring from technological obsolescence and lack of aggressive management, and I had to figure out what to do next. Experience had taught me that software was a much better business than hardware, so that's what I decided to do. . . .

A very different world

In 1982, it was far from clear which way things were going to go. Had the economy continued to deteriorate, had the world drifted toward further polarisation and confrontation, or had the microcomputing market become increasingly fragmented and unable to evolve industry standards able to sustain a viable application market, we would not be here today plotting how best to steer Autodesk through the uncertain times ahead. Indeed, we might not be here at all.

But, in 1982, the downtrends in so many things economic, political, and technological began to turn up ever so subtly. Slowly things began to improve. Confidence returned, technology resumed its exponential growth, and Autodesk never lost the position we had staked out for ourselves in 1982—the world leader in computer aided design. Time passes. Things change. People adapt. Suppose I had included the following paragraphs as the inspirational closer of the original Working Paper for the organisation of Autodesk?

> Who knows what the future may hold? Perhaps, as the 1980s pass into the last decade of this

century, we'll see a world very different from the dark world of today and, if we draw our plans carefully and strive to make them real, the fruit of our labours may have grown into an enterprise that leads its industry into that new decade.

Small changes add up. Reality eventually will out. Today, we're trying to scrape together enough money to give our partnership a six month lease on life: enough for one good shot at success. If we fail, we go back to our jobs and chalk it off to experience. If we succeed.... If we succeed....

The year is 1990. After much strife and suffering, chaos and confusion, we finally ended up naming the company "Autodesk". Every founder has made a million dollars or more from his original investment; those with the greatest confidence in the company have reaped the greatest rewards. Every two cents invested in Autodesk stock has grown to more than sixty dollars. One of Autodesk's products has become the overwhelming global leader in its market—indeed, it created the market, and has spawned an entire industry of related products.

Today, we're worrying about filling our treasury. In 1990, I see our company sitting on a hoard of cash exceeding $140 million and viewing that very evidence of success as a problem: "what to do with all the money." It's a nice problem to have.

Our company's success is world-wide, and what a different world it will be in 1990. The Berlin Wall will have been smashed to rubble by sledgehammers and Eastern Europe will have emerged from the aftermath of World War II to become what it was before—simply part of Europe. The Soviet Union will have a democratically elected parliament, debating the means and pace of transition from central planning to a market economy. More than half of the Soviet republics will be in various stages of secession from the Union. Our company, today dreaming of rolling out our first products in a trade show in the United States, will have organised successful trade shows in Moscow and Prague. And, as the tide of liberation sweeps the world, we'll remain in the vanguard by planning a show in Capetown for 1991.[325]

A very different world.... The Dow Jones Industrials have flirted with 3000 before pulling back; the U.S. national debt moves from trillion to trillion almost unnoticed, and Japan becomes the manufacturing power in the world. And in computing? Well, no end is in sight. Today, we're constrained by a 64K memory limit. In 1990, people will talk disdainfully of machines still limited by a "640K barrier". The typical machine used to run our products will run between one hundred and one thousand times faster than the machines we're using today, and an entire market will have been created for hardware products developed specifically to accelerate the performance of our software. And looking forward from 1990, none of these trends appear limited by any fundamental constraints. Technological visionaries foresee computing at the molecular level—with the very stuff of life—and a time where individual computers and databases will coalesce into a common information pool providing access to the collected wisdom of humanity. And our company will be at the forefront of realising both of these goals.

Would you have joined a company organised by such an obviously raving lunatic?

[325] Man, talk about prophetic! I basically put this in as a joke, since Autodesk had recently narrowly decided against a boycott of South Africa when threatened with being blacklisted by a U.S. group. Things change over the years, and sure enough on August 7th through 10th 1991, the first ACAD Expo Africa was held, midway between Johannesburg and Pretoria (OK, I got the city wrong), and I represented Autodesk at the show. On November 25, 1992 Autodesk opened its first subsidiary on the African continent, Autodesk Ltd. (Africa) in Sandton, a suburb north of Johannesburg; Bill Gordon was the first Area Manager for Africa.

Max Q

Autodesk's trajectory in the first nine years has been steep, yet smooth. Autodesk's entire history to date has coincided with the longest period of uninterrupted economic expansion in this century. In the personal computing market, the IBM PC defined an open, extensible standard that was adopted worldwide as the desktop computing environment of choice. Cumulative change in computing has been revolutionary, but incremental in occurrence; Autodesk has not had to weather the transition from one standard to another as happened when machines of the IBM PC generation supplanted the CP/M Z-80 machines that went before.

Today, however, the future looks unsettled in every direction. There is every reason to believe that the calm and comprehensible environment in which we've built Autodesk may become turbulent and chaotic for a while. Autodesk enters this turbulence at high velocity, continuing to accelerate. It will be a period of maximum dynamic pressure on our company—Max Q. Autodesk's financial strength, market position, technological leadership, control of the channels of distribution, support for a wide variety of hardware and software platforms, and global diversification equip us to ride out this period of turbulence and emerge intact: strengthened, perhaps, with regard to competitors ill-prepared or unable to anticipate events that affect their plans.

But regardless of Autodesk's fundamental strengths, if even a minority of the events I suggest below come to pass, managing Autodesk in the next several years will be much more difficult than in the past. Hard choices will be required, and decisions with enormous consequences will have to be made with little assurance that they are correct.

In this paper, I'll try to survey some of the forms of turbulence that Autodesk may encounter in this period of Max Q, suggest alternatives for dealing with possible events that may transpire, and examine how Autodesk can emerge stronger and better positioned for the next period of growth which will follow, as surely as the changing of the seasons.

Modern Problems

Having been called "Chicken Little" in the *San Francisco Chronicle*, I will remain consistent with the reputation given me by that august publication and enumerate some of the challenges we may be forced to deal with in the next two years.

In order to make my recommendations for dealing with these complicated and interrelated contingencies more clear, I'll first run through all the problems, then discuss what we might do about them in a separate section. Please don't get too depressed by this view of the future; Autodesk is in superb shape to ride out the turbulence I see ahead.

Recession

Even before the invasion of Kuwait and its aftermath in the financial markets, the economies of most industrialised countries were teetering on the verge of recession. After a prolonged economic expansion fueled by unprecedented growth of debt, strains were becoming apparent and were reflected in anemic GNP growth and dismal performance by the majority of stocks (excepting a few, highly capitalised, blue chips).

With $40 oil, interest rates kept high to prevent a further slide in the dollar, the drain of a half trillion dollar

bail-out of what is laughably called the "thrift industry," an incipient credit crunch induced by banks cutting back lending as their real-estate loans go sour, worrisome inflation numbers beginning to appear, a nasty bear market under way in both Wall Street and Tokyo, and new taxes thrown on top of the whole mess, the near term outlook is absolutely awful.

The consensus has been that there would probably be a recession, but that it would be a mild one. Unless I'm missing some fundamental fact or misinterpreting the data, what's coming looks awfully like a replay of 1974–1975 to me. And in that bone-crusher most companies weren't buried in debt as they are today.

This evaluation is based entirely on economic fundamentals. If a large scale, costly, and protracted war should erupt, the severity of the recession will be magnified enormously.

Bear market

I believe the overall trend of the stock market turned down well before the Gulf crisis erupted. Fundamentally, stock prices reflect the prospects for corporate earnings, and given the economic circumstances I listed above, it's hard to be optimistic about the near term outlook for most companies. Once a bear market truly takes hold, it begins to feed on itself: headlines about plunging markets impair confidence, spending falls, sales reflect this, profits erode or turn into losses, dividends are cut, capital spending plans are scaled back, etc., etc., until the whole process ends in a downside spasm of despair and exhaustion, forming the base for the next advance.

Bear markets typically play out in three phases. In the first phase, stocks drift lower for no apparent reason. Most investors believe the sell-off is a normal consolidation before a further advance. In the second phase, stocks are falling for obvious reasons—companies are reporting lower earnings or losses, and many are skipping or reducing their dividends. In the final phase, stocks fall because people are dumping them to raise cash. Nobody wants to own stocks any more. This period of panic liquidation and exhaustion sets the stage for the next bull market (which occurs in three complementary phases). Today we're moving from the first to the second phase of this bear market. There isn't anything like enough fear around for this to be the final days of a bear market. In the recommendations section I'll discuss specific targets we can use as benchmarks for making stock-related decisions in this market.

Real estate crash

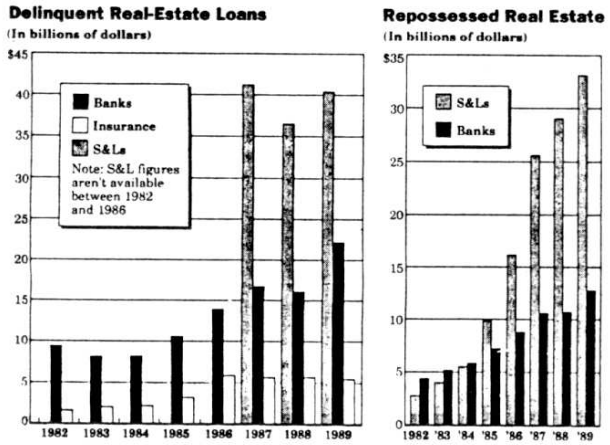
For years, doomsayers have been foretelling the end of the absurd spiral in real estate prices. The argument for a downturn in the residential market is compelling; when most first-time buyers are priced entirely out of the market, you don't just need a greater fool to sustain the market, you need an infusion of new, wealthier, greater fools to buy. And wealthy fools are, as usual, in short supply.

Finally, these forecasts seem to be coming true. Even the California market, one of the last holdouts, now seems to be succumbing to the collapse in prices that appeared first in Texas and New England. While Autodesk's fortunes are tied only tenuously to the real estate market (although clearly the AEC sector of our business will be hard hit by a downturn in construction), the consequences of a sustained drop in real estate will interact with other financial events and magnify the economic problems.

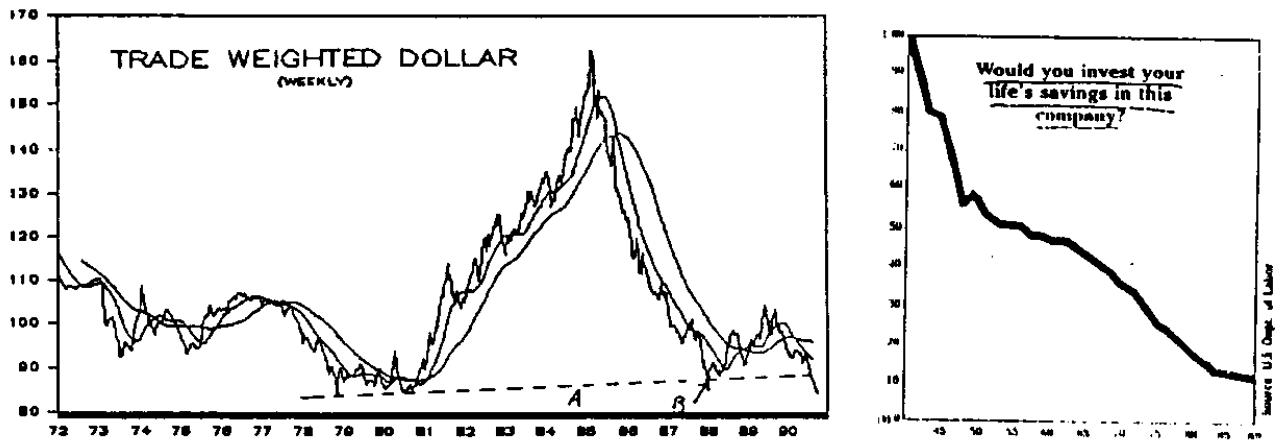
Most of the wealth in the United States is in the form of real estate. As housing prices have risen, this has contributed not only to a general perception of prosperity and confidence, it has contributed real liquidity as well through the ubiquitous home equity loans. If this comes to an end, not only are people going to have less

money to spend and less inclination to spend it, an enormous number of loans are going to start to look very iffy. As reserves are built against these loans, less money will be available for other purposes and this will contribute to the business downturn.

I believe we may be in the relatively early stages of a liquidation of some of the excesses in the real estate market, particularly in California. Even if things get no worse here than they are today in Texas, we have a long way to go. I think that this situation bears careful consideration before contemplating any major real estate purchase commitments.



Dollar collapse



Yes, the dollar has crashed again. This has wiped out all the gains of the 1980s and reduces the real value of our liquid assets which are held overwhelmingly in dollars. More significantly, perhaps, it raises expectations of inflation and forecloses the option of revving up the economy by cutting interest rates.

Debt liquidation

Underlying most of the current troubles in the financial markets is the mountain of debt piled up in the 1980s. Government, corporations, and individuals have all leveraged themselves to the hilt, and as the economy slows, more and more borrowers may find themselves unable to service their debt.

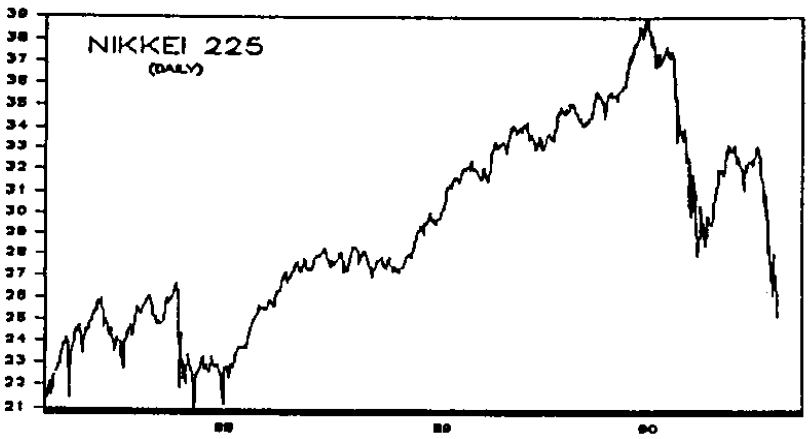
So far, we've seen a rolling, industry by industry, liquidation of bad debt: oil drillers, Texas commercial real estate, junk bonds, savings and loans. Although each has contributed to the demise of the next and the overall tab continues to rise as each domino falls, so far the individual problems have been contained and prevented from leading to a general credit crunch that could cause widespread bankruptcies. Obviously it will be harder to avoid such an outcome if a deep recession occurs.

The debt to equity ratio of so many companies may give the coming recession a very different character from those of the past. In a company with minimal debt, a business slowdown leads to layoffs, reduced inventory, postponing or canceling capital investment plans, etc. Each of these contributes to the slowdown of other businesses and spreads throughout the economy. But a highly leveraged company is very likely to hit the wall by defaulting on its debt very early in a recession. If the debtor takes the obvious step of filing Chapter 11, the business continues to operate while a reorganisation plan is developed and implemented. Chapter 11 reorganisations give a high priority to the company's operating facilities and employment, while wiping out equity investment and either writing down debt or transforming it into equity in some manner.

Although this shrinks the money supply by wiping out equity and debt investments, the overall consequences may be better for the economy than protracted shrinking of a company, slowly squeezing its suppliers and the communities in which it operates. Or, it may be disastrously bad, leading to a chain-reaction of bankruptcies and illiquidity in the debt markets just when debt financing is essential to recovery. I don't know which way it will go; the point is that it's a new situation so behaviour in previous recessions can't provide much guidance as to what will happen this time. It bears thought and careful watching.

In times of tight credit, a cash rich, debt free company such as Autodesk is king. If we enter a period of accelerated debt liquidation (and imagine how insane it would have sounded five or six years ago to talk about a half-trillion dollar collapse of the entire savings and loan industry), Autodesk should pay even closer attention to the quality of the investments in which its cash is placed. It is far better in uncertain times to forgo a fraction of a percent of interest than to lose one's money at the very moment it's most valuable from a strategic standpoint.

Japan



The stock market and real estate situation in the U.S. is mirrored in Japan. However, prices in the Japanese stock and real estate markets have risen to levels utterly disconnected from any conventional measure of fundamental value. This suggests that once they begin to fall, as is now happening, they may have a long, long way to go

on the downside.[326]

This is worrisome not only from the standpoint of the Japanese economy but also for its consequences for the U.S. Japan has been a major exporter of capital in the last decade—reinvesting its trade surpluses and providing liquidity by purchasing debt in the U.S. A collapse in the Japanese market could cause this source of capital to dry up as it is diverted to domestic Japanese needs, and this could accelerate the problems in the U.S.

Dealer devastation

I've been worried about the health of our domestic reseller channel for over three years. Many of the dealers who sell the bulk of our products today, in an expanding economy, are marginally profitable or losing money. As I pointed out last March in the memo "Are we bankrupting our dealers?" (copy attached), Autodesk's pricing policies may have contributed to this situation. If the economy contracts even a modest 10%, the probability is extremely high that many of our dealers would be forced out of business or into other sectors of the market.

If this happens, and Autodesk is not prepared to cope with it, it could be a catastrophe of the greatest magnitude for our company. Reseller distribution has been absolutely central to Autodesk's entire business concept since 1983. It is our dominance of the reseller channel that has made it so difficult for competitors to assail our market share. (The fact that we react so quickly to any attempt by competitors to recruit our dealers is evidence that we recognise this fact.) The very composition of our products and the structure of our company have been built around the dealer channel. Our promotional campaigns and sales literature reflect it. Devolving installation support, customisation, consulting on hardware configurations, selection and integration of application packages, informal training, and, most importantly, post-sale support, upon our dealers has been essential in maintaining Autodesk's very high profit margins over so many years. In a very real sense, we have offloaded many of the traditional cost centres of a CAD company onto our reseller organisations. The price we've paid for doing this is the risk we now run if those businesses begin to fail.

Unfortunately, the dealer viability crisis not a problem which Autodesk has much leverage to solve, as far as I can see. If, as I believe the market has demonstrated unambiguously, the value of AutoCAD to a customer is between $2,000 and $2,500, then further price increases will only further consume the resellers' margin in forced discounts, reduce overall volume, and contribute to market share erosion in favour of lower-priced alternative products.[327]

Although Autodesk has contributed to its dealers' woes, the fundamental problem is not of our making. In the early days of AutoCAD, it took an experienced dealer simply to obtain the odd collection of hardware that was required to lash up a machine suitable for running AutoCAD, get it all to work together, and tweak the system so it delivered acceptable performance. Many of the peripherals required by CAD users, notably professional quality digitisers and large drafting plotters, had essentially no retail distribution channel at all. And, early AutoCAD was no great shakes when it came to ease of installation or the initial learning curve for a new user to become productive.

In such an environment the AutoCAD dealer added real value that was readily perceived by the customer. This is usually the case in emerging markets—it closely paralleled the development, a few years earlier, of

[326]The Nikkei has been up and down since then, but at this writing (August 1992) is in a full-scale free-fall which has taken it down to 15,000. Whether the consequences of this mentioned in the next paragraph will come to pass and, if so, how severe the aftermath will be remains to be seen.

[327]This is not to say that a price hike won't generate an immediate revenue bulge. I'm speaking here of medium-term consequences which will begin to take hold four months to a year after a price increase occurs.

businesses which configured and sold word processing systems based on CP/M machines and WordStar. The eventual demise of that business provides insight into what may be happening now in our reseller channel.

As the market has matured and the price-performance of computers and peripherals has skyrocketed, a CAD system is no longer a odd collection of exotic hardware that can be assembled only by a skilled technician. Instead, today one can order everything needed to assemble a high-end CAD system over the telephone, toll-free, and expect that when the boxes arrive it will take no more than a careful reading of the instructions to get everything working. Architectures such as the Macintosh and SPARCstation where component integration has been carefully thought out contribute to this trend.

In addition, a customer who previously turned to his local dealer as the only source of knowledge about AutoCAD can now read any one of dozens of books about the product, attend courses at a local school, call a 900 number support service, read an article in *Cadalyst* or *Cadence*, post a question on CompuServe, or turn to a fellow AutoCAD user for assistance either at a user group meeting or informally—our very success has engendered a proliferation of alternatives to the services originally provided only by dealers.

I wish these fundamentals were different or that I could find a fix for the problem. Our dealers have worked very hard for us for a very long time, and their efforts have been largely responsible for our success. But if the dealer channel is becoming nonviable, it is incumbent on Autodesk's management to anticipate that eventuality and have plans in place to cope with it should it occur.

Bad quarter

Sooner or later we're going to have a down quarter.[328] If you'd asked me the odds in 1985 that we'd make it more than five years from the IPO without blowing a single quarter, I'd have said "no better than one in twenty." It is an absolute miracle that we've been able to maintain our unbroken streak in the face of late product introductions, hardware lock crises, stop-ship orders, and all the other individual tribulations that disappear into the totals on the financial statement.

However, we have operated so far in an expanding economy. If the economy turns down, our dealer channel begins to encounter severe problems, or we're forced to cut our price to maintain market share, our unbroken streak will come to an end. This is inevitable, and is virtually certain in a serious recession.[329]

While failing to meet expectations for a quarter might seem trivial alongside the other dire circumstances in this gallery, it will have a detrimental effect on morale at the very time when morale and sound judgement are most important.[331] The currently-circulating jokes about Oracle demonstrate how rapidly a company can go from darling to dogshit. If Autodesk appears headed for a bad quarter, it is important that the company, the analysts, and the shareholders be prepared for the aftermath so that one piece of bad news doesn't end up contributing to more.

[328] See page 718 for what happened when we did.

[329] In a severe recession, naturally, analysts will reduce their estimates for all companies, Autodesk included. Nevertheless, it is still likely that Autodesk's results would at some point "disappoint" the analysts as well as fail to set new sales and profit records.[330]

[331] No matter how vivid your imagination, it's still worse when it really happens. See page 718.

Technological confusion

Ever since 1982, the general shape of the PC market has been relatively simple. The IBM PC defined a hardware standard; MS-DOS became the software standard, and applications battled for supremacy in the huge market that adhered to these standards. Although it was far from clear at the time, other architectures such as the Lisa, Unix workstations, Amigas, the Atari ST, and even the Macintosh were largely sideshows compared to the 90% of the market represented by the PC. The assumption that the PC would continue to develop along the obvious lines of faster processors, larger discs, more memory, better networks, and higher resolution graphics has been correct over Autodesk's entire history. Autodesk has profited enormously from this progress.

Recently, however, the future of the PC market has become very confusing to try to project. Since Autodesk, as an application vendor, is forced to choose which architectures will receive the attention of our limited technical, marketing, and promotional resources, uncertainty about the direction of the market complicates our job immensely. Worse, the same confusion that vexes us causes even more bafflement among our dealers and their customers. This lack of direction is bad for everybody—at the minimum it causes customers to defer purchase decisions until things sort themselves out. A customer forced to choose may end up with an incompatible system and waste his money and time on something that ends up being bypassed by the market.

What I'm basically talking about here is OS/2. I guess every decade or so IBM has to show the new kids in the industry who think they know how to screw things up what a débâcle of truly Wagnerian scope looks like. With the able assistance of Microsoft, uncertainty and confusion have been unleashed in the PC market to such an extent that people cannot even figure out which *vendor* is responsible for what product, no less when they might be able to buy a solution to the problems of existing PC users. Consider this: today there are more than forty million computers in place that share a common hardware and software architecture. Yet at this moment there is *no clearly-defined path* for the future migration of these machines which represent a collective investment of many billions of dollars.

What's a user to do? Struggle along with MS-DOS? Move to Windows and buy a whole new set of applications compatible with it? Choose products built with DOS extenders, then fight the mutual incompatibilities among them? Wait for OS/2...but which OS/2? Certainly not the awful one you can buy today, but the one coming from IBM or the one coming from Microsoft? But wait, Microsoft says they're developing a "portable" OS/2 that will run on non-Intel hardware. So are we going to end up scrapping our '386 machines for MIPS or Sparc chips? Crazy—but what's this news about Compaq narrowing their choice for a CPU vendor for their next generation to MIPS and Sun, Intel being notably absent? Chuck the whole thing and get a Macintosh? But a SPARCstation is less than $5,000 and a whole lot faster. And then there's the other side of IBM participating in OSF and promising true cross-platform application portability and yet another wonderful standard environment.

It's entirely possible that all of this disarray in the market may result in nothing more serious than an interlude in which the users continue to buy vanilla '386 and '486 DOS machines while they wait for the dust to settle and a clear direction to emerge. If that's the case then Autodesk need only be careful to avoid squandering too much of our development resources on each of the contending architectures until a winner becomes apparent. But if rampant confusion seems to be hurting the market for our products and further endangering our dealers, Autodesk must be prepared to provide guidance as to what we believe is the best path for our customers, even if it means abandoning some of our usual even-handedness with regard to hardware and operating system alternatives.

Autodesk confronts an uncertain future

After this litany of dismaying facts, we can comfort ourselves with the knowledge that Autodesk could scarcely be in a stronger position to face whatever tomorrow may bring. Below I'll discuss specific recommendations for actions Autodesk should take to prepare for and react to the various contingencies I've discussed. What surprised me most in performing this analysis was how little any of these recommendations have to do with any particular view of the future. In no way do these recommendations constitute a "bet on bad times." In fact, the most important thing of all is that regardless of what happens, Autodesk must remain focused on the most important aspects of its business: developing the best products, making them widely available at prices people can afford, innovating in distribution, support, and training, making the funds we commit to promotion go as far as possible, and preserving the financial strength that allows us to take a long term view of the market and our position within it.

Worse than any specific blunder would be for Autodesk, in the aftermath of a downturn in sales and profits brought about by a contracting economy as opposed to any errors on our part, to start flailing around with gimmicks and lose the confidence of our many constituencies. The higher the seas, the more important is the need for a steady hand on the tiller.

Many of the recommendations I discuss below have been raised before, and some may already be in various stages of implementation. I believe that even if an unexpected burst of prosperity startles everybody and the dark clouds fade away, Autodesk will be better able to cope with good times once these policies are in place.

Conserve cash

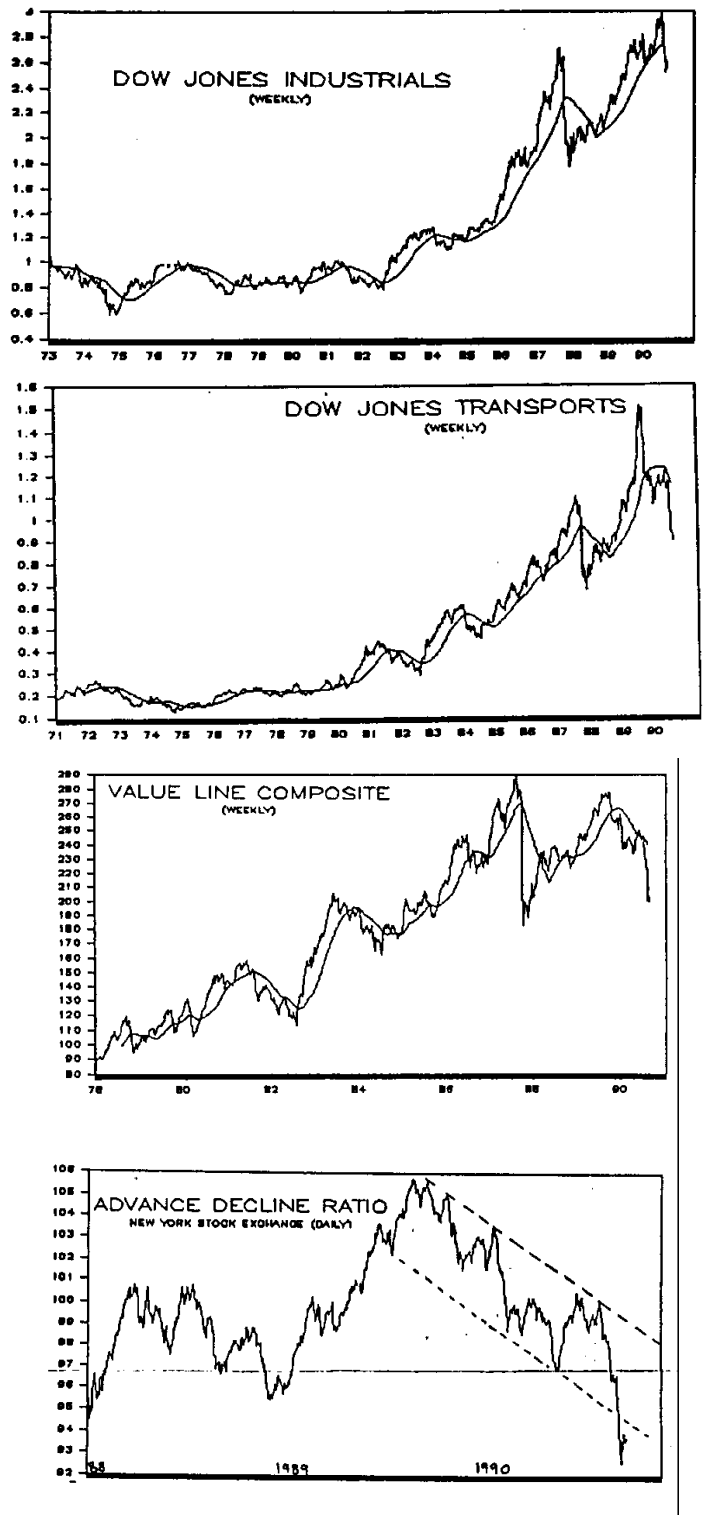
Cash in the bank is always nice to have, but especially in a credit crunch. I agree that Autodesk presently has more cash than we need from the standpoint of survival, but I'd be disinclined to reduce the cash hoard at this time. If we muddle through the present economic problems, we can always bring down the cash once things are more settled. But if we do get a credit crunch and sharp recession, we'll be able to generate substantial earnings from our cash (and apply our high tech company P/E to it, as discussed on page 16 of "The New Technological Corporation").

In addition, there's no better time to be a buyer than at the bottom. If we see a repeat of the 1974 recession, it's very likely we might be able to make some very attractive acquisitions and/or investments at bargain basement prices as the bear market reaches the point of panic liquidation. If this kind of situation begins to emerge, we should devote some effort to scanning the horizon for technologies that fit and which will prosper when recovery begins.

Consequently, I recommend that we do nothing to reduce our cash position at this time.

Don't buy back stock at current levels

One of the bargain basement equity investments we can make with our cash is, of course, buying back shares of our own stock. By reducing the number of shares outstanding, we multiply the earnings per share from constant earnings, making each remaining share more valuable.



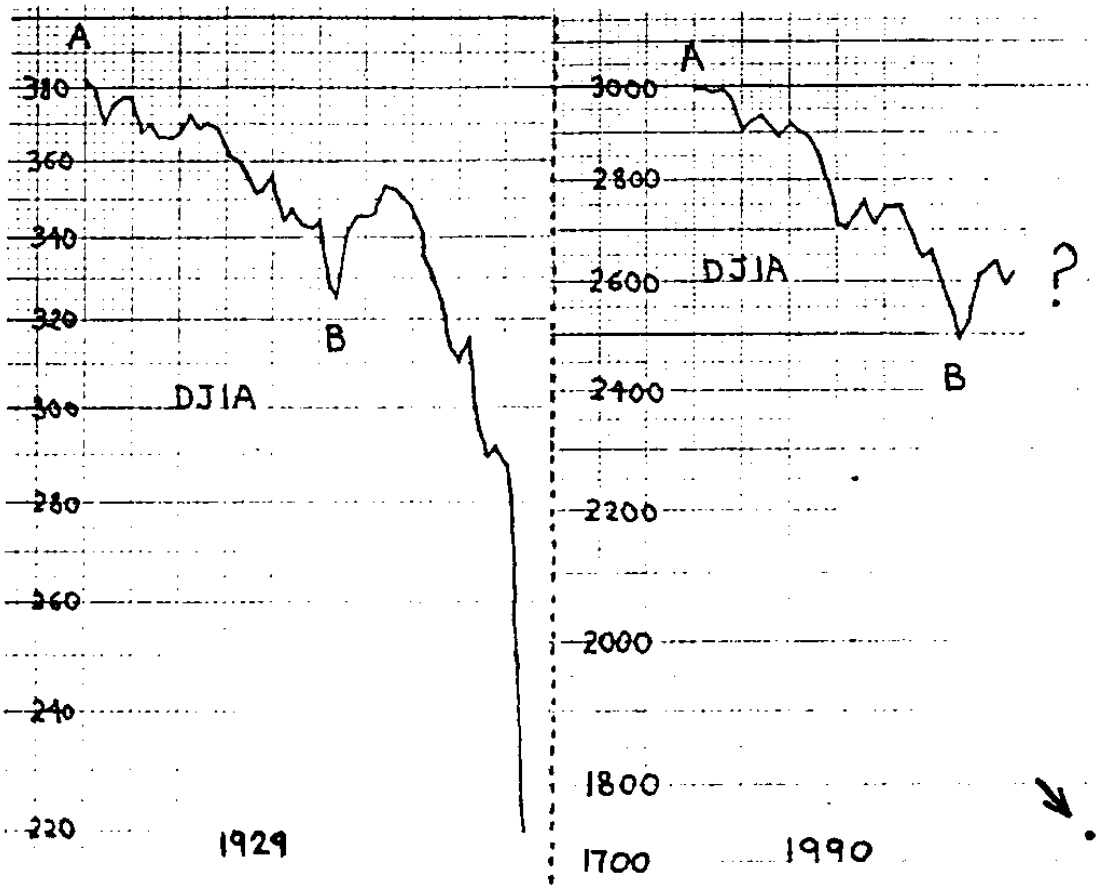
I have nothing against such a strategy, but I don't think now is the time to do it. Every measure I consider reliable tells me that we are in a primary bear market which began several months ago. Strength in the Dow has masked an ongoing massacre in the wider market. The Value Line Composite Index, for example, which comes far closer to reflecting an average stock portfolio, has utterly collapsed and is heading for its 1987 crash low. The advance-decline ratio (breadth), an unweighted measure of all stocks, has fallen to levels not seen

since 1985, when the Dow was in the 1300s.

One guideline for analysing very long term stock market action that's been reliable for most of this century has been the dividend yield on the Dow Jones Industrials. At bull market tops, such as in 1987 and the recent record highs, the yield usually drops to 3% or a little less. At bear market bottoms, such as 1974 and 1982, the Dow yield rises to 6% or more. I believe we're currently in a down-cycle which started with the yield well below 3% and will continue until real fear and panic liquidation grips the market. The Dow yield is presently in the 4% range, so I think we have quite a way to go both numerically and psychologically. If no dividend cuts were to occur, a Dow yield of 6% would require the Dow to fall to the 1750 range. In a recession, of course, dividend decreases are the norm, so an even lower target would be expected.

Interestingly, two other unrelated measures suggest a target in the same range. Bear markets have a tendency to erase half the gains of the preceding bull market. If you look at the entire 15 year bull market that carried to Dow from 577 in 1974 to 3000 in 1990, the halfway point is in the 1780s. Finally, the crash low of 1987 was 1738, with a secondary low of 1766 recorded in December of 1987. These lows have not been revisited in the subsequent years. A chartist (and yes, experience has taught me not to disregard this tool for comprehending mass human behaviour as reflected by the markets) would say there is "strong support" in the mid 1700s.

Then there is this perspective....



Even if you discount all of this analysis as utter mumbo-jumbo, I'd still urge deferring any stock buy-back on the simple grounds that it's wise to conserve cash at a time when there are so many economic uncertainties. If you believe I may be right about the bear market and expect it to follow the historical pattern of past bear

markets, then we should keep the cash in the money bin until the bear brings prices down.

Bear markets, once underway, tend to accelerate and exceed even the most carefully calculated downside targets. The best indication of the end of a bear market is when the market refuses to go down further even in the face of the bleakest of news. Consequently, numerical (or, some would say, numerological) targets, however calculated, are no substitute for informed judgement based on consideration of all relevant factors. That said, if Autodesk stock is today around $45 and the Dow is in the vicinity of 2500, given the β of our stock and the historical tendency of smaller stocks to fall further in a bear market, I'd expect us to be able to repurchase our stock between $20 and $30 per share at a bear market bottom.[332] This is a case where being wrong doesn't cost very much. If our stock starts soaring again, we wouldn't want to repurchase it but then there wouldn't be any reason to! As long as a downtrend is in effect, I'd be inclined to continue to ride it down so as to get the most stock for the funds we devote to the repurchase.

Don't buy real estate

I don't think it would be wise to make any large real estate purchases at this time (in particular, buying a campus site and the initial buildings there). Every indication is that the real estate market is in a downtrend, with no immediate end in sight. Given the overall debt crisis and the amount of highly-leveraged speculative real estate that could come on the market in a real credit crunch, we may be seeing only the beginning of a prolonged bloodletting in California real estate. Even if things get no worse than they presently are in Texas, that's still quite a way from California's present situation.

Conserving cash is the primary motivation. Not wishing to buy into a bear market in real estate is a secondary consideration. Even if you reject both of these rationales, however, I still think we'd be well served by leasing rather than buying simply on the standpoint that Autodesk is a software company, not a real estate company. Why dilute our business focus by starting up a property management operation, even if for our own exclusive use? If the argument in favour of buying is that we would make money by owning our facilities, then I'd suggest that rationale deserves even closer scrutiny because it counsels speculation in California commercial property as the justification for commitment of Autodesk's funds.

I'm not against speculation. But I believe in speculating in things that are likely to go up. I don't think California real estate in 1990 would be on anybody's hot list.[333]

Diversify out of the dollar

This was a good idea when I recommended it mid-1988. It would have been seen, in retrospect, as an enormously shrewd strategic move had we implemented it as we'd discussed around the time the dollar hit its peak in 1989. It's still the wisest course, in my opinion, as the dollar plunges to new daily lows.

This is not a speculative strategy designed to profit from exchange rate instabilities. It is not a bet against the future of the United States. It is simply a prudent and appropriate deployment of the liquid assets of a

[332] Bingo. In January and February of 1992, Autodesk repurchased about a million shares of its stock. The first lot of 210,000 shares was bought at an average price of $31.61 per share and the second lot, 752,000 shares, at $28.37. Of course it wasn't a "bear market bottom."

[333] Autodesk subsequently decided to buy a plot of land on the former Hamilton Air Force Base in Novato, California for $22.5 million and build a "campus" there. This deal fell apart in early 1992.

multinational company which derives close to half its sales and a majority of its recent growth from markets outside the United States.

Once we implement effective currency diversification, we can *ignore* all the headlines about the dollar plunging and soaring. We'll know that Autodesk's ability to compete in each of its markets is unaffected by such news. As long as an overwhelming percentage of our assets are concentrated in one currency, we're held hostage to the monetary policy of the central bank that controls it. Given the experience of the last two decades, this is simply unwise.

Continue to think internationally

Autodesk has done an excellent job in balancing concern for domestic markets with the attention the headquarters organisation must give to the health of the company's operations around the world. We must continue to do this, even if problems in the domestic market threaten to divert our attentions from the global performance of the company.

We've performed superbly overseas, but we should bear in mind that on a GNP basis, if our position in global markets equaled that in the United States, U.S. sales would be only 33% of global sales, not about half. The relative technological, commercial, and political importance of the United States has been eroding for more than forty years, and there's no reason to believe this will end at any time in the near future—particularly in a multi-polar world increasingly focused on competitiveness in a global market economy. Autodesk should deploy its efforts in various markets in proportion to their present and estimated future revenue prospects, not based on arbitrary divisions into territories.

Our products must continue to reflect the international nature of our market. Medium term goals to this end should include, in my opinion, the ability for any Autodesk customer in the world to order and obtain in a timely fashion any of the language editions of our products (in other words, I should be able to order a Czech language AutoCAD from my local Autodesk dealer in Osaka), and to support within AutoCAD and include with the standard product fonts that support all human languages in which engineering drawings are prepared.

Throttle back immediate growth plans

The last thing we want to do is follow the usual high-tech company pattern of continuing to grow right into the crunch, then have to painfully reverse course and cut back. Now's the time to put a lid on hiring, shrink as appropriate by attrition, and lean down the organisation for uncertain times. It's always much easier to turn the spigot back on than mop up the floor after the tub's overflowed.

Based on my own personal leading indicator: the percentage of the job opening bulletin board in 2320/1 covered by paper, I suspect that such a policy is already in effect. Very wise—and appropriate until domestic sales growth resumes.

One of the reasons I'm uneasy about the campus proposal stems from the same source. I'm not sure where we expect the growth to come from that justifies the employment figures the campus is intended to accommodate. And how soon do we have to decide, particularly in a collapsing real estate market that appears to favour the patient buyer?

Prepare alternative distribution plans

Nobody hopes for a collapse of our reseller channel. *But what if it happens?* Management must manage. Even if you assign a small probability to the scenario I've described which could leave us in short order without a channel for moving our products, wouldn't it be wise to assign a commensurately small working group to explore alternatives should that come to pass? What I envision is the preparation, discussion, and adoption of several contingency plans for reacting to anticipated problems in the dealer community. What is important is that these plans be sufficiently specific and well thought out so one could be implemented merely by disseminating it throughout the company and giving instructions to execute it.

I do not wish to prescribe the strategies such plans might recommend. They could range from utter abandonment of the reseller channel and a transition to direct-mail and telemarketing with a greatly expanded direct support operation to a program of equity investment in key resellers combined with pricing changes intended to guarantee the viability of the surviving distribution organisations.

Thinking about this today, before the problems become severe, will not exacerbate the situation any more than drawing up a will hastens your demise. It's far better to worry about a potential crisis and then discard a contingency plan as unneeded than to be forced to react without adequate time for reflection on the consequences of various courses of action.

Provide guidance to resellers and customers

In a market confused by OS/2 vs. Windows, RISC vs. CISC, MCA vs. EISA, and so tiresomely on, it may be necessary for Autodesk to forgo our traditional neutrality and provide some real guidance to our user and reseller communities about what makes the most sense.

Clearly, we wouldn't want to do this in a way that impaired our relationships with a wide variety of vendors, but we not only bear a responsibility to our customers to provide an honest evaluation of their alternatives in assembling a system to run our software, it's in our own best interest to have the majority of our customers equipped with optimal hardware and software configurations for our products.

If no better alternative presents itself, and you believe this problem is as serious as I suspect it is, I'd recommend that we create another Autodesk Distinguished Fellow of a heavy-duty programmer and charge that fella with going out into the real world and telling the truth about hardware and software for doing CAD.

Do Windows

One specific consequence of the confusion in the PC market appears clear: users are moving to Microsoft Windows in droves. It provides many of the apparent benefits of OS/2 and the Macintosh at a low marginal cost. We've been caught without an AutoCAD for Windows, and we should remedy this omission as soon as possible. My understanding is that no technical barrier prevents us from immediately starting a Windows port with the intent of shipping a Release 11 Windows version within the next 12 months. The Windows AutoCAD would run in 80386 protected mode under Windows, as Mathematica presently does, and would have performance comparable to that of AutoCAD 386.

AutoCAD as an open system

With the re-architecting of the internals of AutoCAD anticipated for Release 12 (the OOPS project), Autodesk will be in a position to take a bold step which, if successful, may ensure the preeminence of AutoCAD for the next quarter century, greatly accelerate the pace of AutoCAD development, and establish a new paradigm for the relationship between a PC software vendor and its customers which our competitors will find difficult to emulate.

I'm talking about making the source code for AutoCAD available, and before you stop reading, let me explain the reasons for such a move as well as the means I've come up with for testing the concept without incurring any substantial risk.

Why distribute source code? Because we're greedy, and we'll make more money that way. Consider Unix. Unix source code has always been available. In the early days, when Unix was primarily used within universities, AT&T's licensing policies made source available to most users of the system. As a consequence, *an entire generation of computer science students came to think of all operating systems in terms of Unix*. I believe that the dominance of Unix in the workstation market today is directly traceable to that policy. Tens of thousands of people learned Unix as their first system, and learned it in depth as only access to the source can enable. They learned how to adapt Unix for other machines and applications. By thinking of all operating systems within the Unix paradigm, they demoted those other systems to second class status and made it difficult for any of them to assail the position Unix had achieved, not just in the market, but in the very minds of the decision makers.

We can do the same thing with CAD. Today, AutoCAD has become synonymous with CAD, but only from the perspective of the user. By making source available, the very implementation of AutoCAD will become the paradigm for how CAD systems are built. Universities will use AutoCAD to teach computer graphics software development, and a generation will come to think of graphics software in the language we've defined—AutoCAD. But that's not all.

For those fanatically dedicated users will also be *doing things* to their AutoCAD source code. They'll be implementing extensions, replacing antiquated algorithms with the latest concepts from current research, investigating new ideas in user interfaces, and they'll make all of these developments available to us. Why? Well, because that's part of the source license but, more importantly, because they want the recognition that comes from having a vendor adopt their work. If this seems like fantasy, please recall that most software developers at Autodesk are using workstations that run an operating system developed by AT&T, provided in source form to its customers, one of whom, the University of California at Berkeley, extended it into the operating system of choice for workstations and handed those changes back to AT&T.

All of the founders of Autodesk learned their trade by working on software provided in source code form by its vendors. I think any of them will readily confirm that such experience tends to make one view all subsequent systems in terms of the one you've mastered at the source level. Within five years, a major role of our software development department may be evaluating various user-developed extensions to AutoCAD and integrating them into an ever-changing product.

But isn't this giving away the store? Not at all. First, if somebody wants to steal AutoCAD, starting with the source code is a pretty idiotic way to go about it, compared to just copying the distribution discs. There's no need to worry about disclosing any deep, dark proprietary information either, since AutoCAD doesn't really contain any. All of the algorithms used in AutoCAD are available in the open literature and for the most part are used by all CAD systems. In any case, a creative user can always reverse-engineer technology without source code, as the customers who've decoded the **.DWG** file format have demonstrated.

But what about serialisation, hardware lock code, and so on? Release 12 holds the answers to these concerns. The OOPS mechanism allows code to be written in the same manner regardless of whether it is within what we now call the "AutoCAD core" or outside in an "application." By defining the core as a very small module, provided without source, containing the security-related code, and most of the rest of AutoCAD as outside the core, we can provide source to all the portions of AutoCAD a legitimate user might want to examine or modify without compromising the security we have today in the core code. There is no efficiency penalty in structuring the program in this manner, so it's largely a question of where we choose to draw the line.

Do we really want users to see how awful some parts of AutoCAD are? Absolutely! Then some wild-eyed kid at some backwater college will rewrite the **OFFSET** command so it really works right and mail it in. This really happens. Trust me.

But what if I'm wrong? How can we justify taking such a risk? Simple, try it out on a little piece first. In Release 12 we have the opportunity to separate IGES and dimensioning, two of the most-modified sections of AutoCAD, portions that have accounted for an endless series of user requests for change, and experiment with source licensing them. Clearly, nobody is going to be able to knock off AutoCAD by having access to these pieces. Let's go ahead and make IGES and dimensioning applications at the OOPS level (something that makes sense in any case, from purely technical considerations within the product), and then dip our toe in the water by introducing source licenses for them at, say, $1500 for both. The educational price would be something like $50. (My long term goal for a full AutoCAD source license price would be equal to the runtime license. For example, $3500 for the binaries and another $3500 for the source. All source licensing sales and administration would be direct, and hence all sales would be at full list price.)

Yes, indeed, this could generate additional revenue, couldn't it?[334]

Anticipate technological change

There's one good thing you can say about recessions. They sure do make hardware cheap! The 1974–75 recession was a major contributor to the subsequent personal computer explosion in a very simple yet often overlooked way: at the end of that downturn, the price of logic circuits per gate had dropped from dollars to pennies. Projects that previously required the financial resources of a substantial company were now within the purview of individual mad scientists in drafty basements. I know. I was one.

The semiconductor business is an analytical economist's dream. It has a huge capital spending component and long lead times, a large intellectual capital contribution that's related to R&D spending and the ability to translate it into marketable products, and close to zero material costs. These fundamentals have interesting consequences in a recession.

The R&D costs for the products you're currently making are already sunk and can be recovered only by selling more products. The cost of operating the production line is dominated by amortisation of capital equipment already in place and direct labour—both unrelated to the volume of products produced. Incremental improvements in yield reduce costs and are sought on that basis alone. The cumulative effect of these fundamentals is very much like a replicator that's swallowed its shut-off switch. With the factory accounting for a constant fixed cost every month, all incentives are toward increasing output. Increased output drives the unit price down,

[334] Given the runaway risk-aversion rampant at Autodesk during this era, I didn't really expect this recommendation would be acted upon, and it wasn't. On June 2, 1993 Bill Gates announced precisely such a program under which Microsoft will license source code for Windows NT to major research centres including Brown University, Carnegie Mellon University, Massachusetts Institute of Technology, Stanford University, University of California at Berkeley, University of Washington and Xerox Palo Alto Research Center.

forcing competitors to further streamline their own operations, and so on.

The nature of semiconductor fabrication technology causes memory circuits to lead all others in complexity at constant cost. A software company whose products have been constrained by limited memory for the better part of a decade would be wise to ponder the consequences of a memory price crash. (Or, should no recession eventuate, a slower yet equally inevitable erosion of memory prices.)

When Autodesk was founded, 64K memory was the standard complement in the personal computers of the time. I recall when I first heard that the IBM PC version of AutoCAD would require 128K of RAM. "How profligate," I thought, "Why can't they fit it into 64K like real programmers?"

At that time, 16K RAM chips were giving way to the 64K RAM generation. Since then, we've seen the 64K RAM supplanted by the 256K RAM, and the 256K pass the torch to the 1 Mb chip. Recently, 4 Mb RAM chips have begun to appear in quantity with prototypes of its successors, the 16 and 64 Mb chips already described in research papers at the various solid state circuit conferences.

As we sit and struggle with MS-DOS and its one megabyte address space limitation, it's worth contemplating the implications of a world that's adopted the 4 Mb or 16 Mb RAM as the standard memory component. Most of the microprocessors in use today have a 32 bit memory bus: each memory access reads or writes four bytes at once. Hence, when that memory is composed of chips of a given capacity measured in bits, the memory expansion increment in bytes is four times the chip capacity in bits.

Think about it. As 4 Mb RAM chips achieve price parity and begin to displace the 1 Mb generation, the *minimum* memory configuration of a 32 bit bus processor such as an 80386 or 80486 will be 16 megabytes, and memory expansion will be in increments of 16 megabytes. When the 16 megabyte chips supplant the 4 Mb chips, memory will start at 64 megabytes and grow in steps of that size.

Memory on this scale truly changes everything. CAD moves from an application that strains the limit of every resource on the system to a modest user of abundant computing resources. Anticipating the widespread availability of computing power in this class (which will happen regardless of the economic environment, but perhaps later rather than sooner), Autodesk should devote resources within the Advanced Technology division to defining the design tools made possible by such hardware configurations and undertaking their development in order to deliver them as soon as the required hardware becomes available to our customers.

Summary of recommendations

- Conserve cash.
- Defer stock buyback until lower prices.
- Avoid real estate purchases.
- Implement currency diversification for cash portfolio.
- Continue internationalisation of products and operations.
- Limit near-term growth plans.
- Prepare contingency plans for disruption of distribution channels.
- Consider recommending hardware and software migration path for AutoCAD customers.

- Implement AutoCAD Release 11 for Microsoft Windows 3.0 as soon as possible.
- Cautiously explore AutoCAD source licensing through a low-risk pilot program.
- Estimate mainstream computer configurations in the 1995 period and begin to plan products around them.

Conclusion

Never since our company was formed has the immediate future of the economic and political system within which Autodesk operates been so uncertain. Credible, rational alternatives for the next several years range from global peace and prosperity as democracy and free markets sweep the globe to financial panic, economic collapse, and World War IV. Whatever may come, Autodesk is well-prepared. We have the financial strength to withstand hard times, the cautious and prudent approach to risk taking needed to protect that strength, a commanding position in an enduring market that is central to industrial development, and the international and industrial diversification to mitigate localised or sector-specific downturns.

What Autodesk must do to survive difficult economic times is largely the same things we do to take advantage of periods of rapid growth—making the best products and providing the best service to our customers. Considering the recommendations in this paper and implementing those judged to be wise should further insulate Autodesk from unpleasant surprises and allow us to remain focused on the truly important things. The principles that have brought Autodesk the strength to face the future with equanimity will guide us through that future to even greater success.

Information Letter 14

Information Letter 14 is, arguably, the most widely-distributed internal memo in the recent history of the computer industry. In November 1990, I took my second six-week leave which, combined with accumulated vacation days, amounted to three months away from the fury of software development. I relaxed (a little), read (a lot), wrote a book "The Hacker's Diet", and spent a lot of time evaluating the numerous software products which were being launched for Windows 3.0. I quickly reached the conclusion that the issues about which I'd been concerned when I penned my August 1988 "Technological Leadership" memo to Al Green (see page 430) were not only unresolved, but in many cases the situation had deteriorated, both internally within Autodesk and vis-à-vis the competition, the evolution of the market, and the expectations of customers.

I wrote Information Letter 14 for presentation to senior management, to spur them from what I perceived as a dangerously complacent view of the software business. With all the turbulence that has ensued since 1991, it's easy to forget that the problems and risks I warned of in Information Letter 14 were little-perceived around Autodesk prior to its appearance. After its appearance, one Autodesk director said of Information Letter 14 "some of it was unfair and a little exaggerated." Management, who quickly moved to minimise the significance of Information Letter 14, apparently persuaded investors that all was well—whatever impact the release of Information Letter 14 had on the stock was limited to a week or so, after which the stock went to new highs.

Legend has it that Information Letter 14 was "released into the company's E-mail system" as the "ultimate flame mail" or similar nonsense intended to paint me as an irresponsible leader of some kind of underground "cabal" of "Core programmers". For example, and for a good laugh, see the front-page "profile" of Autodesk in May 28, 1992 issue of The Wall Street Journal, written by hatchet-man Greg Zachary.[335] *What actually happened was that, as I always do, after drafting an early version of IL 14, I circulated draft copies to a few people with whom I'd been discussing these issues over a period of time, to see if they felt the document was complete, accurate, and fair. Unfortunately, I failed to adequately impress upon one of these reviewers the confidentiality of the memo, which was intended first for senior management and then, only later, for general distribution. As a result, copies of the early draft began to circulate within the company and outside, which forced me to quickly issue an official IL14 with at least some of the shortcomings of the draft corrected.*

The runaway replication of the unofficial draft meant that the copy most people read was not the final document presented here, in the form I delivered it to senior management on April 1, 1991. The differences were not great, but I consider this version the more accurate

[335] See page 730 for the story behind that article.

in expressing the issues facing the company than the early draft. Information Letter 14 inspired Bill Gates, who described it as "brilliantly written and incredibly insightful", to write his own Issues and Strategy "crisis letter" in which he said that "By talking about how a large company slows down, fails to invest enough and loses sight of what is important, and by using Microsoft as an example of how to do some things correctly he manages to touch on a lot of what's right and wrong with Microsoft today."

The Final Days

Autodesk, Inc. Information Letter # 14

by John Walker

Revision 24 — April 1, 1991

Here on the level sand
Between the sea and land,
What shall I build or write
Against the fall of night?
Tell me what runes to grave
That hold the bursting wave,
Or bastions to design
For longer date than mine.
—A. E. Housman, 1936

Prologue: The Quaint Eighties

I've noticed something odd over the last few months. Whenever I read something written between 1982 and 1988, or reflect upon those years, they seem increasingly distant, foreign, almost *quaint*. Quaint in the sense the Eisenhower years seemed by 1968, or the earnest hopes of the early sixties from the depths of the mid-seventies. Who would have imagined a few years ago that in the first months of 1991 the news would be filled with a war in which an alliance of the Soviet Union, Syria, Britain, France, Egypt, and the United States used high-tech weapons to flatten an Arab country, of the reorganisation of Europe around a united Germany exporting, among other things, rubble from the Berlin Wall and curios of the departing Red Army, and of a collapsing Soviet Union which even Russia *(Russia!)* was considering abandoning, careening into a crisis of unknowable magnitude and consequences, spurring sober observers to fear "a nuclear Beirut."

If the pace of change in the world seems breathtakingly fast and ever-accelerating, developments in our own software industry are even more rapid and revolutionary. Often it seems like the pressing concerns of six or twelve months ago are no more relevant to our current priorities than the Wars of the Spanish Succession or the controversy over N-rays. Compounded exponential growth is thrilling to experience and pays well, but it demands of those who would prosper by it the ability to make ever larger adaptations with less and less time to prepare.

When a company ceases to change at the rate demanded by the industry it exists within, it finds itself rapidly left behind. Before long, its customers discover products of competitors that better meet their needs. As market share slips, sales fall, and earnings decline, the management of the standstill company asks, "What's happening? We're still doing all the things we used to do."

Surely they are, but that's no longer enough. Times have changed but they did not. Increasingly their company and its products seem like relics from the past, almost... quaint.

Introduction

I am writing to you because I am deeply concerned for the future of our company.[336] Autodesk has been successful over the last nine years because it quickly adapted to the changes in the marketplace for its products. I believe we are embarking on another period of rapid evolution in personal computer software, one fully as significant as that ushered in by the IBM PC in 1982. That product defined the personal computer software industry as we know it. The era we're now entering holds unparalleled opportunity for companies with foresight to anticipate the transitions and position themselves to benefit, creativity to build the next generation of products, aggressive management driven to get the job done and bring it to the customers by energetic marketing, and the financial strength to accomplish all these tasks in times of economic uncertainty.

Autodesk possesses all the prerequisites to lead the next generation of the PC industry, yet it seems to have become stuck in the past, mired in bureaucracy, paralysed by unwarranted caution, and to have lost the edge of rapid and responsive product development and aggressive marketing and promotion on which the success of AutoCAD was founded.

Not only has Autodesk failed to bring the new products it needs to the market, it is allowing AutoCAD, our flagship product and the source of essentially all our revenue, to become dangerously antiquated and undermarketed to an extent that is virtually unique for a product generating sales in excess of $200 million a year.

Just as the rapid changes now underway hold great opportunity for those who exploit them, they imperil companies which fail to adapt. Among software industry leaders prior to the IBM PC, only Microsoft remained in the forefront in the 1980s. The software battlefield is littered with the corpses of companies who had a great success with one product and then neglected that product until it was eventually supplanted by a new, more imaginative product. CP/M, VisiCalc, and WordStar in the past... Lotus 1-2-3 today... and tomorrow? I believe that unless Autodesk acts immediately and decisively, effecting a rapid and comprehensive top-to-bottom change in what the company believes possible and how it goes about accomplishing its goals, AutoCAD will suffer this same fate, destroying with it all we have worked for so long and so hard, extinguishing forever the promise and opportunity that Autodesk still holds in its hands.

Background

In the early days of Autodesk, I wrote status reports on the progress of the company for employees and shareholders (who were, at the time, the same people). These papers chronicled the triumphs and disappointments, the crises and their resolutions which are part and parcel of building a company. It's been years since I brought my view of the company to you in this way; since I removed myself from management in 1988 I've been concerned primarily with software development and identifying technical directions in which the company should move, not commenting on where it was going or how effectively it was getting there.

But now I feel compelled to speak out. I believe that our company is entering a time of great peril combined with unparalleled opportunity. Regrettably, I do not believe that Autodesk's management is positioning the

[336] This opening was a deliberate parallel with the first sentence of my June 1983 "Crisis Letter" (see page 185).

company to emerge from this period stronger and better equipped for future growth. Indeed, it is my opinion that their current policies place at risk everything we have achieved since 1982.

First a few words about me and my relationship to the company. As you probably know, I initiated the organisation of Autodesk, was president of the company from its inception through 1986, and chairman until 1988. Since I relinquished the rôle of chairman, I have had no involvement whatsoever in the general management of the company. On occasion, management has sought my opinion on various matters, in the same manner they consult others with relevant experience and insights, and from time to time I have volunteered my opinion on various issues, both verbally and in writing. My view has always been one of many inputs weighed by management when reaching their decisions. Over the years I have agreed with many of their choices and disagreed with some, but all in all I felt our company was in good hands. In any case, I never doubted our senior management was doing a better job of running the company than *I* ever did when I was involved more directly.

Some people mistakenly believe I still "call the shots" in some covert fashion; that despite my repeated and sincere expressions to the contrary, somehow management either rubber-stamps my decisions or grants me veto power over their judgements. Nothing could be further from the truth; were that the case, I would certainly not be bringing these concerns to you today in this fashion. I have raised these issues that trouble me so deeply with senior management repeatedly and forcefully. My premises and the conclusions I draw from them have not been disputed. In fact, on numerous occasions, I was told action would be forthcoming to implement many of my recommendations. But nothing has happened. Well, something has happened: time has passed. And as the months and years go by, the difficulty of refitting Autodesk for the realities of a new era in the software market increases as its importance grows. Inaction in the face of a changing market and world is the chief cause of my concern for Autodesk's future. I see in it the same somnambulistic plodding to the precipice that preceded the demise of so many former leaders in the software market.

Also, let me state unambiguously that regardless of the sentiments I express herein and the direct manner in which I characteristically state my opinions, I have no desire whatsoever to see Autodesk's management removed from their jobs or to resume any rôle in management myself. What I want is for them to *act:* to act in the same way managements of other companies of similar size in comparable industries facing similar challenges act—to do what is *necessary*, not what they've come to believe is possible; to make the difficult choices they are paid to make and put the company back on the path to further growth and success.[337] The decisions will not be easy, their implementation will not be simple, nor will the process be devoid of pain. But the alternatives are all much worse.

As you may know, for reasons largely unrelated to the matters I discuss herein, I have decided to permanently leave the United States. It was my plan to continue my work in software development at Autodesk's new software development centre in Neuchâtel Switzerland. That is still my intent, unless Autodesk responds to this message by silencing the messenger.[338]

[337] At this time, I had no doubt in the ability of senior management to accomplish the kinds of changes I envisioned herein. I was confident that once the issues were understood, action to address them could and would follow. It was only later, when as summer gave way to fall and the only visible action that had occurred was a "re-organisation" that I began to doubt the ability of senior management to face the challenges before Autodesk.

[338] As I'd expected, Information Letter 14 led to no personal confrontation with anybody in the senior management or, for that matter elsewhere. The aftermath involved an intense discussion of strategy and tactics, which was conducted in an entirely amicable and professional matter. I must say, though, that I did wonder if somebody Upstairs didn't want me to finish Information Letter 14. Here's a description of what happened to me the night before I delivered the "draft" that got away, from an E-mail I sent at 4 **a.m.** to one of the people who was expecting the draft.

> I'll send you a copy of my current screed if/when I get it finished. "If" seems like a real possibility at the moment. Last night, around midnight, as I was putting in some edits and getting ready to finish it off, a mud and water volcano erupted

Finally, in this paper I will largely focus on AutoCAD, as opposed to the company's other products. First, the simple reality is that AutoCAD is where all the money comes from, and therefore developments which threaten it threaten the company. Utterly botching AutoShade, Animator, or Xanadu would be tragic, but would not bring the company down. Allowing AutoCAD to lose its leadership would. Second, the problems that I see afflicting AutoCAD are the same problems faced by the other products. For a number of years I believed that Autodesk's lack of success with new products stemmed from a dangerous fixation on AutoCAD; no other product received the attention necessary to make it a success because its near-term contribution to revenue was swamped by that of AutoCAD. Now, however, I believe that AutoCAD has suffered from the very same neglect, both in the product development resources committed to it and especially in marketing and sales aimed at expanding its market. The inaction that led to the lackluster performance of AutoSketch and AutoShade after their introduction is today threatening to destroy AutoCAD. The redirection of the company which must occur to rescue AutoCAD will benefit all our products.

It isn't possible to discuss the state of Autodesk and AutoCAD or to adequately describe the competitive risks I believe threaten our company and its products without speaking frankly of the shortcomings of AutoCAD, contrasting it with the products of other companies, and, in pointing out Autodesk's vulnerabilities, providing a roadmap a competitor could use to mount an assault against Autodesk. I have thought long and hard about the risks of bringing such information to the attention of a wide audience; surely, given the first- and second-hand distribution of this document, copies will fall into the hands of the press and competitors. Given that the author is a founder of the company, excerpts may be used in the short term to embarrass Autodesk or to promote competitive products. I have concluded, though, that these risks are unavoidable consequences of placing the issues I discuss here on Autodesk's agenda. If Autodesk acts as I believe it must, it will quickly render impotent competition based on its prior weaknesses. If Autodesk doesn't move to remedy these shortcomings, they will soon (if not already) be sufficiently obvious that competitors won't need me to point them out, nor my guidance to draw plans to exploit them.

---

in the corner of my computer room. Apparently...

1. The automatic (solid state) water level switch on the sump pump failed causing,
2. The sump pump to fail to activate causing,
3. Groundwater to fill the sump with mud causing,
4. The pump, when it did finally trigger to fill the outlet hose with mud and rocks causing,
5. The pump's thermal protection to trigger causing,
6. Failure of the pump causing,
7. Runaway accumulation of groundwater behind the retaining wall causing,
8. Old face-full.

Other than about 3 hours of sleep last night between manually bailing out the sump beneath the house and waiting for the hardware store to open, I have been working on this continuously from then until about 1:45 **a.m.** today. Naturally, the full story was nowhere as obvious as I painted it above—particularly the plugged output pipe which really threw me for quite a while. Of course, now I get to clean up the mess the original disaster created plus all the muddy footprints I made trying to fix the problem.

Fighting the sump pump (which included such joys as digging out about a trash can full of mud and rocks with an entrenching tool, which was the only thing small enough to use in the space beneath the house) kept me from being able to collect the brush and pile it by the roadside during the day when it wasn't raining. This is Muir Beach fuel abatement week, when you have to get rid of the dead brush (the tree chipper will start to run in about 4 hours), before the fire department comes around to ticket you if you don't clean up. So, I spent the hour from 2 to 3 hauling dead branches in the dark, in the rain and (briefly) hail.

How have things been going for you?

"The Game Has Changed"

Throughout the proposal, organisation, and early operation of Autodesk, my constant theme, repeated until I'm sure everybody was thoroughly sick of hearing it, was "the game has changed."[339] From the perspective of 1990, the original concept and mode of operation of Autodesk seems hopelessly naïve. It would certainly be so today, were anybody foolish enough to think they could enter what is now a mature industry in so amateurish a way.

But in 1982, I used the phrase "the game has changed" to shock people into realising that even then the stakes were rapidly rising and that to build a successful software company would require funds, commitment, professionalism, and risks far in excess of what previously characterised the personal computer business. Sometimes people forget that personal computers were already six years old when the IBM PC was introduced, and that several companies had grown to $10 million per year or more manufacturing CP/M, Apple II, and other early machines. What had been, in 1977, a game into which anybody with a bright idea and a soldering iron could jump in had, by 1982, become a serious business in which millions were made and lost.

The first PC software fortunes had already been made. CP/M from Digital Research, MicroPro's WordStar, and Visicorp's VisiCalc dominated the software landscape to such an extent that some believed no opportunities remained to found new mass-market software companies.

Yet today, none of those companies commands a substantial position in the market. What happened? The game changed, but they did not. As the game changed, the stakes to stay in it grew enormously and those companies, the former leaders, failed to summon the resources they needed and the courage to deploy them. What one day looked like an utter, unassailable monopoly fully as secure as AutoCAD's grip upon the CAD market evaporated within months at the hands of competitors with products that better served the customers in the new environment. Times have changed; clear the screen; turn the page.

When the IBM PC appeared, the expectations of software customers rose rapidly. Software purchasers would no longer settle for a disc with a handwritten label, a five page manual photocopied from a dot-matrix original, or unreliability of any kind. The standards of quality, professionalism, presentation, and support all rapidly escalated, and those companies who survived were those who realised the bar had been raised and did what was necessary to continue to clear it. Indeed, the great successes of the early IBM PC era: Microsoft, Lotus, and Ashton-Tate, were the very companies that raised expectations through their own products. Since that time, standards have continued to rise and the struggle for supremacy in the mainstream business applications: word processing, spreadsheets, and databases, has largely been contested by increasing product quality, functionality, and customer service.

When major shifts occur in user expectations, dominant hardware and software platforms, and channels of distribution, companies which fail to anticipate these changes and/or react to them once they are underway are supplanted by competitors with more foresight and willingness to act. The displacement of Digital Research by Microsoft, of VisiCalc by Lotus, and the current eclipse of 1-2-3 at the hands of Microsoft Excel are all examples of this process.

[339] See, for example, the original Working Paper, page 14.

The Game Has Changed Again

It is my belief that AutoCAD as a product, and Autodesk as a company, is poised to lose its market leadership in precisely this manner. Further, I believe this event is overdue and that Autodesk is living on borrowed time provided only by the absence, as yet, of an effective competitive attack aimed at Autodesk's true weaknesses—one coherent with the emerging characteristics of the software market. Today Autodesk is king of the mountain, but it is poised precariously, waiting to be pushed off by any company that seizes the opportunity and acts decisively. One of the largest unappreciated factors in Autodesk's success has been the poor strategy and half-hearted, incompetent execution that characterised most of our competitors in the past. But betting the future of our company on this continuing for another decade is foolish, a needless prescription for disaster.

During the years when AutoCAD pioneered the market for PC CAD, Autodesk constantly innovated in means of distribution, support, training, promotion, applications—every aspect that contributed to the present success of AutoCAD. Today, Autodesk seems frozen in the past, as if the clock stopped sometime in 1987 or 1988. There seems a cargo-cult-like belief that merely going through the motions that worked so well before will guarantee similar success in the future. But we did those things because they were right for the market several years ago, not today. The game is changing again, and Autodesk shows no signs of adapting to the newly emerging era.

Modern Times

What are the characteristics of the software market that is emerging in the 1990s? Sometimes we are so clever in our analysis that we overlook important points simply because they are so obvious. The products that are building new markets today and becoming the new stars of the software firmament are:

- Big,
- Cheap,
- Widely available, and
- Closely integrated with the platform.

Let's look at each of these characteristics in turn.

Big—more software in the box

The most fundamental characteristic of modern software is that it is *extensive*. Just as the first IBM PC applications dwarfed their 64K CP/M predecessors, modern software exploits the resources of machines with megabytes of main memory and hundreds of megabytes of hard disc: turning the potential latent in that hardware into benefits the user can perceive. This is what I call "the quantity of software in the box," and it is the most obvious metric of software value per dollar spent.

Here are some data points to ponder, showing the amount of software delivered with each of the following products:

Product	Size (Mb)	Executable (Mb)
Dbase III (1985)	0.5	0.13
Lattice C 3.0 (1986)	1.4	0.53
Autodesk Animator	1.6	0.48
Windows Excel 3.0	4.6	1.95
Word for Windows	4.7	1.32
High C 1.6	5.0	2.43
Asymetrix Toolbook	6.5	1.18
AutoShade 386 2.0	7.6	3.74
PowerPoint	7.6	1.30
AutoCAD 386 R11+AME	8.8	4.11
Windows 3.0 SDK	8.9	n/a
3D Studio	9.2	1.32
Borland Turbo C++	9.3	4.14
CorelDRAW!	14.2	3.37
Microsoft C 6.0	14.5	5.62

(These figures were obtained from the size of these products as installed on my Compaq 386. Some products, if installed with different options, can vary substantially in size. For example, when PowerPoint is installed on a system with a Hewlett-Packard LaserJet instead of the PostScript printer I use, it includes special downloadable fonts that increase its size to 18 megabytes.)

The actual executable program, what most folks in software development and consider to be "the product," is a fairly small component of the total software delivered to the customer. Most of the size of modern products comes from what Autodesk dismisses as "support files" and devotes relatively little effort to: fonts, sample documents, help files, on-line tutorials, on-line documentation, clip art, menus, macros, templates, and so on.

But as seen by the user, these components are just as much a part of the product as the executable program. First users appreciate items like on-line hypertext help and documentation. Then they expect them, and soon they demand them. This kind of massive support around the core of an application is becoming a prerequisite for software products, especially those in large, maturing markets.

Another aspect in which modern products are "big" is in what are called "production values" in the movie business; the appearance of the product, documentation, and packaging. I remain a firm believer that, all else being equal, the product that delivers the greatest functionality and performance to the customer will win out in the end. But there's no law of engineering that requires a powerful product to look crude or behave in a less than civilised manner, as if products somehow derived virtue from the software equivalent of exposed screw heads, sharp corners, and chalky grey paint. Simply compare the appearance of AutoCAD with Excel or Word for Windows. Which product looks like it costs $4000?

Let us consider an example, quite close to home. Here is a comparison of AutoCAD with CorelDRAW!, a leading 2D drawing and illustration product for Windows.

	AutoCAD	**CorelDRAW!**
Fonts included	26	150
Clip art	None	6 Mb
Sample drawings	1.6 Mb	1 Mb
Symbol library	143 symbols	3000 symbols
Colour support	256 colours, proprietary	Pantone, RGB, HLS, CMYK
Line types	24 + user-defined	35 + user-defined
Fill patterns	53 vector + user	35 vector + user 49 bitmap + user 42 PostScript + user 8 grey scale linear, radial fountain
Arrow heads	3 + user-defined	87 + user-defined
Transformations	translate, rotate, scale, mirror, extrude	translate, rotate, scale, mirror, extrude, skew, envelope, bilinear, tween
Drawing librarian	None	Included
Formats imported	DXF, IGES	DXF, PCX, TIFF, BMP, AI, EPS, GEM, PIC, HPGL, CGM, PIF, PICT
Formats exported	DXF, IGES, HPGL, EPS	DXF, EPS, WMF, PCX, TIFF, CGM, PIF, GEM, HPGL, AI, PICT, SCODL, VideoShow, WPG, WFN
Scanned image trace	Discontinued	Included
Font import	None	Adobe, Compugraphic, Bitstream, DigiFont, Readable PostScript (PFA), Z-Soft Type Foundry
Font export	None	Adobe, Z-Soft Type Foundry
Printed tutorial	228 page, 7 lessons	100 page, 8 lessons
Video tutorial	None	1 hour, VHS
Technical support	CompuServe	CompuServe, Phone, Fax
User newsletter	No	Yes
List price	$3,995	$595

Now I don't mean to imply by this that there are not many features present in AutoCAD that CorelDRAW! lacks. CorelDRAW! is an illustration tool, not a CAD system. It is totally two-dimensional (other than simulated 3D special effects), lacks dimensioning capabilities, has nothing resembling object snap or other geometry-driven constructions, does not support digitising tablets for precise co-ordinate input, and has no macro language, user defined objects, or other facilities permitting it to serve as the basis of applications.

My point is this: there is nonetheless a large overlap between CAD and illustration. Despite the pretensions of CAD systems to be seen as modeling tools, the end product most CAD systems in the hands of most CAD users is marks on paper. Consequently, a CAD product such as AutoCAD and an illustration product like CorelDRAW! will, of necessity, provide many of the same capabilities. And within that large common area,

among all those features these two products share, CorelDRAW! surpasses AutoCAD in every single one, in some cases overwhelmingly. In appearance and ease of use AutoCAD is totally blown away.

You might be tempted to respond by saying, "Well, nobody in his right mind would buy AutoCAD to do illustration, and AutoCAD's geometric construction, dimensioning, multi-view plotting, and layer control still make it far more productive for CAD, even in 2D." I agree. But you'd be missing the point. Just because a product provides a larger set of more complicated features is no excuse for botching the simpler, basic stuff, for allowing the functions that constitute the meat and potatoes of all drawing to be inferior in scope, shrouded in an opaque, obscure, and antiquated user interface, and incapable of exchanging data with other applications in widely-used, contemporary formats.

I've discussed CorelDRAW! in some detail because, as a drawing tool, many of its features are directly comparable to AutoCAD. I've grown more than a little tired of hearing AutoCAD developers burst out laughing or dismiss as impossible capabilities present in a product that sells for a fraction of AutoCAD's price. That many people aren't aware of such products disturbs me; that some are and don't seem to care I find appalling. CorelDRAW! is not an isolated example; it is typical of the modern generation of applications, especially in the rapidly-expanding Windows sector. These applications, including products such as Microsoft's Excel, Word for Windows, and PowerPoint, embody a breadth in the scope of the software, a depth in the supporting materials supplied with it, and appearance and other fundamental production values that make AutoCAD look, if not amateurish, at least antiquated.

Restricting the comparison among products to executable file size gives lie to a widely-shared misconception about AutoCAD (and oft-cited excuse for its feature shortcomings, slow development cycle, and inflexibility): the claim that it is a "big, complicated program"—one that strains every limit of the personal computer and embodies not only more complexity, but simply more code than other widely-used applications. Perhaps this used to be true, but it isn't any more. As the pace of AutoCAD development has slowed (in my opinion, due to the meager human resources devoted to it), other products have been catching up and, in some cases, surpassing AutoCAD. The executable sizes in the table above include utilities, translators, and in the case of AutoCAD, both AME and AME Lite. Here is a table that compares just the core executables.

Product	**Core Executable (K)**
PowerPoint	859
Word for Windows	894
CorelDRAW!	904
Asymetrix Toolbook	1,176
Windows Excel 3.0	1,254
AutoCAD 386 R11	1,815

While at first glance AutoCAD might appear to have an edge, note that AutoCAD is the only program on the list compiled in 32-bit 386 mode, which is substantially less efficient in code and data space. In addition, the other applications inherit their low-level drawing and user interface management from Windows, whereas AutoCAD includes its own facilities for these functions. In an era when AutoCAD can not only be run on a laptop computer, but *developed* on one, the constraints on its growth appear a matter of priorities rather than technology. The reality is that AutoCAD is, today, a medium-sized application and will, in the future, have to become much larger to keep up.

Cheap

Most current software products sell at price points ranging from one sixth to one eighth the price of AutoCAD. Around Autodesk it's sometimes easy to forget just how high the price of AutoCAD is. Also, there's a tendency to forget that this wasn't always the case. When AutoCAD was introduced, it was priced at $1000, and for much of its history it sold for roughly $2000. This was a premium price, but much closer to the mainstream of contemporary software.

In fact, the combination of AutoCAD's high price and high volume is, to my knowledge, unique in the industry. Software this expensive tends to be semi-custom products or packages that address a small vertical market, not something sold by dealers with an installed base numbering hundreds of thousands.

Any company able to command a premium price should feel gratified; it's the ultimate verdict of the market on the quality and utility of the product in question. However, when the premium approaches an order of magnitude above other products with similar or greater development investment and, at the same time, dealers find it increasingly difficult to sell the product at anything approaching the recommended retail price, it's time to ask the following question.

> "What price point for AutoCAD generates the maximum revenue and profits for Autodesk and its distribution channel?"

I'll skip the refresher course on price elasticity curves from Econ 101. Just recall that beyond a given point raising the price of a product *reduces* revenue by causing volume to decrease. In the longer term, overpricing renders a product vulnerable to lower-priced competition.

In an environment where concerns about grey market distribution, software piracy, health of the dealer channel, faltering sales growth, and worries about margins abound, it would seem wise to revisit the question of AutoCAD's price and ask whether it is consonant with the pricing of software products which will maintain and expand their leadership in the 1990s.

Widely available

Modern products are heavily advertised in a wide variety of media, and are available through a multitude of distribution channels. By contrast, Autodesk is committed to *reducing* the number of outlets where a customer can purchase AutoCAD. Again, one must ask whether this policy, adopted with the goal of protecting the dealer channel which has been responsible for a large part of Autodesk's success in the 1980s, will in the long-term, benefit Autodesk or its dealers.

Our policy, "AutoCAD is sold by authorised dealers" has been in effect for so many years and has been defended with such vehemence that sometimes we forget that we never planned it that way. When we started the company, software distributors and publisher/distributors were seen as the primary channels through which software developers would move their products, with sales occurring either directly by the distributors or through local retail computer stores who bought products at wholesale from them. At the same time, retail chains such as ComputerLand, MicroAge, and later Businessland were negotiating corporate arrangements with software vendors to buy centrally for their stores. Finally, many of the computer companies then jumping in to compete with IBM in the burgeoning PC market, including Texas Instruments, NCR, Victor, Wang, NEC, AT&T, Zenith,

and Fujitsu were building software distribution channels alongside their hardware sales network (recall that this was years before the "clone" market emerged, so special versions of each software product had to be prepared for each of these computers).

As we were gearing up to launch AutoCAD, we were developing contacts in these three primary channels: distributors, retail chains, and computer manufacturers. After we announced the product, only the manufacturers seemed interested. The distributors and chains didn't see enough volume in an odd "vertical market engineering package" to justify carrying it, but many of the manufacturers saw AutoCAD as a product that would showcase the better graphics and greater speed their machines offered compared to the IBM PC.

And then a funny thing happened; the phone started to ring. After the first couple of articles about AutoCAD appeared in the PC press, customers started walking into computer stores and asking for it. The stores called us. We signed them up and called them "Authorised AutoCAD Dealers." As our dealer channel continued to grow, principally through individual dealers taking the initiative to contact us, we continued pursuing the manufacturers with some success and the chains and distributors with next to none.

Finally, a few years later, the chains and distributors started coming to *us*, having discovered that many of their stores had signed up directly with Autodesk as AutoCAD dealers. We looked long and hard at the deals they were proposing and did business with some of them, but essentially what they were looking for was a cut on the AutoCAD volume their customers were already selling. We didn't see much benefit in this either for Autodesk or for the dealers who were already selling our product so, in most cases, we declined such distribution deals. (The whole story of the twists and turns as Autodesk evolved its reseller strategy is far more complicated than I can relate here. I've tried to capture the main flavour of it, and hope those whose favourite milestones I've omitted will understand.)

By 1985, a structure much like the present one was in place, with Authorised Dealers selling most of the product, a Fortune 500 program getting underway, and a thriving grey market in AutoCAD that we were trying to stamp out in every way we could manage. And there it has pretty much stayed until now, with relatively minor adjustments through time.

The association of AutoCAD with dealers wasn't something we planned. It just happened, to the great benefit of both Autodesk and our dealers. In retrospect we can see why. In the early days of AutoCAD, just collecting the pieces of hardware needed to run AutoCAD, getting them all to work together, installing AutoCAD on the system and tweaking it to deliver acceptable performance on machines that ran anywhere from 10 to 50 times slower that the typical AutoCAD platform today took a great deal of knowledge and no small amount of work. The drafting plotters that can be bought by calling a toll-free number today were, in those days, sold as specialty items by Hewlett-Packard offices; retail distribution for such hardware had never been contemplated. So it was also for large digitising tablets, high resolution graphics boards, and large monitors.

In order to assemble a working AutoCAD system from scratch a user would have had to become, in a real sense, a computer expert. Far better, especially for a person seeking only increased productivity in drawing, to pay a dealer to put all the pieces together, shake it down into a working tool, and install it along with training to bring the user up to speed. All these tasks the dealer did for the user constituted the "value added" by the dealer, for which the customer paid when he purchased AutoCAD and the hardware at retail, rather than the dealer price.

To a lesser extent, the same could have been said about all other PC applications at the time. Today it might seem absurd to need professional help getting a word processing system running, but only by people who've forgotten some of the horrors that were sold as printers in the early 1980s, or who think of a hard disc as

something you can rely on day in and day out. Those days, computer stores helped ordinary people, unskilled in the strange ways of computers, put computers to work in their homes, offices, and workshops.

Then, as the years passed, computers improved. Not only did the absolute price of computers fall while their performance grew, they became more reliable and manufacturers learned to tailor them better to the needs of specific target markets. In addition, the IBM PC clone emerged as an industry standard architecture. This eliminated many of the compatibility problems that bedeviled the industry previously, especially in software. Now it really was possible to take a computer out of the box, load up WordStar or Lotus 1-2-3, and get right to work.

Inevitably, it wasn't long before customers began to ask themselves, "If all it takes is opening the box and plugging it in, why am I paying this guy at the computer store a thousand bucks to do it?" And soon the first headlines about computer stores going out of business began to appear in the industry press. The customer shopping for a PC or software could readily compare the prices quoted by the mail-order merchants who advertised in the back of all the computer magazines with the stickers at the computer store and, since the products involved had become commodities with reputations based upon their manufacturer's position in the industry rather than the recommendation of a local dealer, saw no reason to pay the premium demanded by the local dealer.

Over the last few years the local computer store, as envisioned in the early 1980s and common in the middle of the decade, has largely vanished. Computers are still sold locally, but frequently much in the same manner as televisions and other electronic appliances, with less markup for the reseller. With constant competition from nationwide mail-order distributors, there's little room for the local retailer to increase his price. Software distribution has changed as well. If hardware has become a commodity, so even more has software. A dealer can add value by unpacking a computer, installing the operating system, adding memory chips, and so on, but each copy of Excel 3.0 is just like every other. Since they're interchangeable, and software installation these days usually consists of "stick the little square thing in the slot and type **`A:SETUP`**," there isn't any reason to pay a penny more than necessary for the product. Consequently, the prices charged for software by direct marketers, discount retailers such as Egghead, and other volume channels are remarkably similar and represent a small margin compared to that of a dealer selling at retail a few years ago.

Once the value added by a reseller begins to disappear, what does a manufacturer gain by restricting the distribution of his product to those resellers? Clearly, withholding the product from mass distribution protects the resellers and helps to maintain the dominance of the product within that channel. By preserving a local sales and service network, the need for direct customer support is reduced, lowering the manufacturer's overhead. Finally, a dealer network, properly supported and directed, can act as a nationwide sales organisation for the manufacturer—one that doesn't come out of his marketing and sales budget.

These are powerful arguments, and Autodesk's success has demonstrated the importance of local presence through dealers. But, as with every other aspect of our business, we must periodically inquire as to the health of our dealer network, and whether Autodesk and the dealers who sell its products can continue to prosper in the coming years as we have in the past.

Here I find serious causes for concern. Compared to the typical desktop computer, an AutoCAD machine is bigger, more complicated, and harder to install and optimise, and the same can be said for AutoCAD compared to most other software. But, just as the passage of time and the evolution of the industry eliminated the need for special skills to get a simple PC running, today they are doing the same for CAD. One can turn to an advertisement and order, with a single toll-free telephone call, a ready-to-run CAD system composed of nationally marketed and serviced components, a system one can fully expect to work as soon as it is plugged

in. What then is the value added by an AutoCAD dealer?

I think the shrinking margin between the price at which Autodesk sells AutoCAD to its dealers and the price dealers are able to obtain for it from customers (the so-called "street price") reflects the perception on the part of AutoCAD buyers that many dealers are doing little more than passing the product through their hands, and thus deserve only a small markup. In such a situation, trying to raise the average retail price by limiting distribution and pursuing "grey marketeers" is like trying to stop the tide with a teaspoon and a sponge; it's setting yourself against the judgement of the market, and it never works, at least not for very long.

A product becomes a commodity when purchasers discover it meets the definition of one: interchangeable, readily available, and sold at approximately the same price by all vendors. Once these conditions are met, there's nothing the manufacturer can do to change the situation. Nothing, that is, that doesn't harm himself. For what possible benefit could there be in making the product harder to obtain, more difficult to put into service, of unpredictable composition, or capriciously priced? Even if a manufacturer succeeded in driving up the retail price by curtailing supply, the effects would, in all likelihood, be short-lived since a sudden, steep rise in the price of a popular product, combined with its disappearance from many channels of distribution would send the clearest possible signal to competitors that here was a market begging for a readily available, more affordable alternative.

It's no secret that many of Autodesk's dealers are encountering difficulties at present, problems that in many cases began well before the current economic downturn. AutoCAD is the last major software product to retain dealer sales as its only channel of distribution. If the problems in the dealer channel are not transient, but instead indicate that dealers can no longer build a profitable business selling AutoCAD, Autodesk could be left in the position of controlling a channel of distribution which was no longer viable. This would leave the field open for other CAD products to establish themselves in the mass market channels where AutoCAD is not for sale.

Closely integrated—the emerging standard platform

Something is happening in our industry, something very important, and we would be wise to recognise its significance and take account of it in our plans. The advent of the IBM PC in 1982 forever changed the nature of the PC software business, even though many software companies didn't realise it at the time. (Autodesk certainly didn't: we spent at least as much effort on CP/M-80 and CP/M-86 machines in the first two years as we did on the IBM). Today, the shape of the industry is again being changed by the emergence of a new standard application platform defined, this time, not by hardware but by software—Microsoft Windows.

Just as there was nothing "new" about the IBM PC in 1982; 8086 and 8088 machines with similar capabilities existed years before, there is nothing at all new about Microsoft Windows except the way the market has embraced it. But that's all that really matters in the long run. Whether you're a Macintosh fanatic, a committed NeXT developer, an OpenLook advocate, or a Silicon Graphics believer doesn't change this fact: more than *eight million* copies of Windows are expected to be sold this year, and that estimate may prove low since rumour has it more than half that number were sold in the first 90 days of 1991.

Whether it's ugly or beautiful, naughty or nice, good or evil, when a product gains that kind of momentum and enlists that many users on its side, software developers had better start paying attention to it. Consider this: foremost among the companies who desperately hoped Windows would never take off was IBM—an outfit known to have some clout in the marketplace and reputed, by those who dislike it, to be able to persuade people to buy almost anything. Yet even in the Fortune 500, the very heart of IBM's market and the segment

it influences most strongly, Windows is spreading like wildfire, brushing away OS/2 as if it had never existed.

Unlike the Macintosh, which has suffered from a premium price, single source, and worries about connectivity, Windows allows a DOS user to upgrade for less than $100 and continue to run all his old software. This both contributes to the rapid adoption of Windows and reinforces users' demand for truly integrated applications; once Windows is on your machine, the distinction between programs that understand the clipboard, system fonts, system printer, and all the other Windows services and those that go *boinggg!!!* and blop out to a dumb old DOS character screen becomes glaringly evident. This makes getting caught out without a Windows version of your program just about the worst possible thing that can happen to a DOS application vendor these days. Just look at the increasingly desperate and strident promotions being unveiled by Lotus to try to maintain sales of 1-2-3 as they feverishly debug their Windows-based reply to Excel.

All of the dynamics that made the Macintosh market so special, that made Macintosh users so unwilling to consider any alternative, are now being ignited by Windows in a market ten times larger. A community of users ten times the size of the Macintosh, Amiga, Sun, and NeXT user base combined is now beginning to discover, albeit in a cruder way, what possessed people to buy those other systems. You don't have to predict the future to see the Windows phenomenon, you need only open your eyes. Consider this: Windows, like the Macintosh, lets you attach a little icon to an application. Users can make their own icons and customise their systems that way. Proud of their artistry, many users upload their spiffiest new icons to CompuServe and other networks so others can share them. The last time I looked there were, sitting on CompuServe, a total of *1.3 megabytes* of Windows icons ready to download. These aren't the applications, just the icons—a total of 1700 of them, including five each for AutoCAD and Generic CADD!

A market ten times the size means ten times the money to be made by application vendors, and if that weren't incentive enough, it's a market that, with each installation, displaces a raw DOS machine. Certainly Windows continues to suffer some technical shortcomings: it only allows 16 bit applications (unless heroic effort is exerted, as in the case of Wolfram Research's *Mathematica*), it is essentially a single-tasking system, and it inherits all the shortcomings of the MS-DOS file system. All of these limitations are, however, scheduled to be remedied over the next two years. Given the importance of Windows to Microsoft's strategy and the resources they commit to such projects, Windows buyers can be reasonably confident the schedule will be met. And even if it isn't, Windows 3.0, as it stands today, is far and away the best environment you can choose without throwing away your DOS hardware, and that's an option most users can't afford, even if they were inclined to.

In addition, there's an effort underway by Microsoft, Compaq, and others to make a machine-independent version of the Windows–OS/2 environment and use it to enter the workstation market. While there's room for many a slip in such glib and grandiose plans, if I were Sun I'd be more than a little worried about the prospect of twenty or thirty companies cranking out RISC machines that ran a user interface already known by twenty or thirty million people, one to which application vendors could port their software to simply by recompiling.

What I'm suggesting is that Windows is a Big Event—the kind of thing that happens every decade or so in our industry that establishes a new baseline from which future evolution builds. Events of this nature reward those who move quickly enough to exploit them and winnow out others whose attention is elsewhere, who underestimate the significance of the change, or cannot react in time. Big Events force those who wish to survive to revisit their strategies and question long-term plans. This process requires flexibility in an organisation which is difficult to maintain after it has grown enormously in size and become set in its ways.

One central and virtually unquestioned tenet of Autodesk's strategy has been "platform independence." That means, with very few and very limited exceptions, we do nothing on any one machine that cannot be done on every machine that runs AutoCAD. This forces us (some would say "gives us a handy excuse to") exclude

support for many facilities on machines like the Macintosh which virtually all other applications, even the least expensive, furnish. The facilities provided to the AutoCAD user become, in a large sense, limited to the least common denominator of those provided by all the machines we support and look, consequently, crude next to applications closely tailored to a specific system. When we feel compelled to address a glaring shortcoming, such as lack of support of system menus and dialogues, we're forced into a much larger project such as Proteus, since our solution must work on every machine and operating system, not just one.

Severely limiting the integration of AutoCAD with various operating environments hasn't hurt us so far, I believe, primarily because the systems that account for the overwhelming percentage of our sales: DOS and 386 DOS, also happen to be the least common denominator in virtually every aspect. Consider all the things we could have done to make AutoCAD faster and easier to use if we hadn't required everything to run in 640K of RAM. Regardless of your opinion of our Macintosh version of AutoCAD, or your view as to how we might have proceeded in that market, the fact is that regardless of whether we succeeded beyond our wildest expectations or failed to sell a single copy of AutoCAD for the Macintosh, our financial results wouldn't have changed much. That doesn't mean we shouldn't have put AutoCAD on the Macintosh, or that we shouldn't continue to strive to better integrate AutoCAD into that environment (I'm the guy who first managed to get AutoCAD running on the Macintosh, you'll recall); all I'm saying is that Apple's market share is simply too small to have much of an effect on our sales (especially when you only count machines capable of running AutoCAD), and rapid expansion of the market generated by AutoCAD is unlikely as long as the Macintosh suffers in price/performance against 386/486 and Sparc machines.

Windows, however, is changing the rules in the very heart of the AutoCAD market. We will, certainly, ship a Windows version of AutoCAD before too many months pass, and we will upgrade that initial product to take advantage of the new versions of Windows now in the pipeline, but I think we have to revisit the level of support we're planning for Windows. Our Windows product will be integrated with Windows roughly to the extent our Macintosh product conforms to the Macintosh environment, which is to say somewhere between "somewhat" and "moderately." Certainly it will be obvious to any user that many rules change when the mouse strays into the AutoCAD window. This situation has almost certainly hurt us in the Macintosh market, but due to the limited size of the market and the fact we didn't have any sales there to begin with, hasn't become a company-wide priority to fix. I believe that a similar failure to comply with ground rules for Windows applications may hurt us severely, and every week that passes without our thinking about how to address this problem adds to the danger.

(I would hope that whatever we do to allow AutoCAD to fit comfortably into Windows will also let us conform as closely on the Macintosh, OpenLook, NeXT, etc. However, if forced to choose between close integration with Windows now and all-platform user interface support in 18 to 24 months, I'd do Windows first and worry about the others afterward.)

The first modern CAD product

I believe that a CAD product with these characteristics: big, cheap, widely available, tightly integrated with its host system, and promoted and marketed in an aggressive manner could, in relatively short order, displace AutoCAD from its current dominance of the CAD market. AutoCAD would not be eliminated, any more than Lotus 1-2-3 has vanished in the face of competition from Excel, but it would be placed in the same difficult position: forced to play catch-up against the more modern product, trying to reverse an erosion of market share against a newer product with momentum on its side.

Autodesk has the capability today, by making a series of decisions at the level of senior management, to bring the first totally modern CAD product to market. Success in this endeavour would protect Autodesk but, most importantly, would position it to resume its growth into new markets and applications: the broadening of the market that accounted for much of our success in the last decade. Failing to take these steps will, in my opinion, leave AutoCAD a sitting duck waiting to be picked off by the first competitor who launches the product that AutoCAD could have been. How long might Autodesk have before that happens? There's no way to know, but betting the company on it not occurring is hardly a prudent strategy, or indeed any strategy at all. Ponder this: in my opinion, the magnitude of work involved in adding AutoCAD's capabilities to an application such as CorelDRAW! is roughly equal to that of adding CorelDRAW!'s facilities to AutoCAD. Further, remember that a competitor is free to target the lucrative heart of the AutoCAD market, not being saddled with the baggage of compatibility with prior releases, unprofitable hardware platforms and operating systems, niche applications, and characteristics of our distribution channel that constrain Autodesk's freedom of action.

Modern Problems

> Problems are only opportunities in working clothes.
> —Henry J. Kaiser

The large scale, value per dollar, wide distribution, and strong marketing of the new generation of software products reflects ongoing changes in the way the software industry goes about its business. As markets have broadened, revenue has grown, and companies have matured from small bands of moonlighting entrepreneurs to members of the Standard&Poor's 500, the scale of the resources they invest both in the development and launch of new products and the ongoing support of already-successful products has grown apace. Notwithstanding the inefficiencies inherent in doing things on a larger scale, the fact remains that a major release of a modern software product simply reflects many more man-hours of labour and dollars of capital investment than products of a few years ago. There is a lot more software in the box, software that meets or exceeds the ever-rising expectations of an increasingly discriminating community of users, because more people worked more hours using better tools to *put it there*, and when that software reaches the market, it is supported by a vigorous, comprehensive, and thoroughly professional promotional campaign, both at the time of introduction and throughout the subsequent product life cycle.

Except at Autodesk, where I believe this process has broken down.

How to lose sales and market share through imagination and hard work[340]

Consider this: open the AutoCAD box, the actual commodity that changes hands when a customer buys our product from our dealer. Take out all the pieces, then go back through the product release notes, the documentation review routing slips, and the like, and make a list of the names of individuals whose work directly appears in that box—the originators of everything that eventually ends up in the hands of the customer. When I do this, I come up with about 15 names. It's no wonder we never seem able to deliver what we would like to on a schedule we can live with.

I do not mean to imply that only 15 people are responsible for AutoCAD, or to disparage in any way the efforts of the much larger number of individuals in quality assurance, development test, product management,

[340]This is play upon the title of my diet book, completed in January 1991, "The Hacker's Diet: How To Lose Weight and Hair Through Stress and Poor Nutrition".

marketing communication, or other aspects of product development: these are just as essential as writing software, producing documentation, and assembling the support materials that constitute a product release. *But they don't wind up in the box!* When the customer takes delivery of the product and unpacks the box, he doesn't see any of those other efforts. He *assumes* adequate resources have been expended to insure the product is reliable, not the least since he parted with such a large wad of cash to acquire it. He *relies* upon the integrity of the product's vendor to protect his investment through upward compatibility and cross-platform data interchange. He *expects* the vendor's financial strength, management resources, and commitment to future development will ensure the company can continue to meet his future needs. But at the moment, once the shrink wrap has been discarded and the floppies copied to the hard disc, all that the user sees, reads, and uses is what was in the box.

Written by about fifteen people.

Is this an appropriate development commitment to a product that is generating on the order of two hundred million dollars per year in sales, a product that commands an overwhelming share of a rapidly growing market?

Modern software has so much in the box because *more people* are working to put it there. Mythical man months and mystical management aphorisms aside, if you stack a development team of 15 people up against a Microsoft-sized project with a hundred or more people directly contributing components that the user will encounter in the product, the tiny team, however bright, however motivated, however hard-working, will always come up short. Especially if the team of 15 people is only allowed to work on the product for a few months per year, conforming to product release schedules proclaimed by accountants (as is the case for Release 12), not by sales, marketing, development, or (perish the thought) the needs of customers.

But doesn't increasing the number of developers, writers, and the like require corresponding increases in the number of supervisors, managers, testers, spec-writers, and everybody else involved with the product? Yes, of course it does. Doesn't that increase costs far beyond even the already large costs of a big development team? Naturally. That is the way the software industry works these days, and it is the investment required of all companies that wish to remain leaders.

But, can Autodesk afford it?

Out of two hundred million dollars a year?

Autodesk has inherited many things from its history, not least of which is the tradition of the "hungry rat," a reputation as a lean, mean competitor that consistently did more with less through imagination and sustained hard work. This mode of operation was essential when there wasn't anything *but* imagination and hard work from which to forge a company. In the absence of market share, distribution channels, reputation, financial capital, or a community of users, you fall back on what remains. This way of doing business took us very far. Indeed, it took us all the way to the last quarter of fiscal 1991 with an unbroken streak of rising sales and earnings. But I fear it can no longer guarantee the future of AutoCAD in an increasingly sophisticated market. I believe it is burning out our best people and ensuring the eventual eclipse of our principal product.

Benign neglect: silence in the marketplace

Not only are development resources committed to AutoCAD inappropriate to its sales, market share, and importance to the company, the visibility of the product in the marketplace, the ultimate result of Autodesk's efforts in marketing and sales, is unseemly for a product of its stature. Autodesk, once renowned for its

innovations in marketing and sales, seems to have settled in recent years for a policy of "All the same things, and less." Divide Autodesk's history into two parts at the halfway point: sometime in 1986. How many new initiatives in marketing and sales have been launched in the latter half?

Autodesk's penchant for abandoning products developed at great cost at the very moment of shipment has long been a source of frustration for me. AutoSketch was the first example of this sorry tradition, and even though our neglect of that product later forced us to spend millions of dollars to buy Generic Software to guard the low end of our market, that didn't keep us from launching both *CA Lab* and *Chaos, The Software* with a marketing budget of essentially zero. In fact, it was only after I offered to pay for the advertisements myself that a small sum was disgorged to advertise *CA Lab* in the issues of *Scientific American* and *Discover* which each devoted a page or more of editorial copy to the product.

The very existence of the Multimedia group is an admission of the neglect for Animator after its hugely successful initial launch. If Autodesk had, in 1983, treated AutoCAD the way it treated Animator and 3D Studio after their introduction, Autodesk would not exist today.

But while any number of reasons can be advanced for neglecting products which some view as distractions from the central business of the company, indulgements of certain influential people, when AutoCAD suffers the very same neglect in the marketplace, the reasons become much more inexplicable and the potential consequences more dire.

But aren't we spending lots of money on marketing? Well, I don't see the budget numbers, but I believe we are—just look at the phone list and make a body count. But the issue isn't how much you *spend*, it's *what comes out;* the equivalent in marketing of measuring development by what goes in the box. This metric reveals the extent to which Autodesk has abandoned AutoCAD, ceding by default the position its preeminence in the market merits to any competitor willing to assail it, leaving the customer perception of the product in the hands of reviewers, analysts, and the authors of books.

Is this an extreme statement? Yes, it is. But I believe it accurately reflects an extremely dangerous situation. I don't understand the logic behind spending $400,000 developing a product like *Chaos*, then allocating essentially zilch for marketing it after all the development cost is sunk. Such a policy makes failure of the product a self-fulfilling prophecy, or at least treats recovering the investment as a crap shoot on users spontaneously stumbling over the product. If you want to save money, *don't develop the product in the first place!* But don't wimp out at the instant the product has a chance to recover its costs and turn a profit.

But I digress. You probably don't care about *Chaos*. Let's look instead at a $20 million investment which has been abandoned in precisely the same way. I am talking about AME—our only entry in the solid modeling market, the very cutting edge of the mechanical engineering sector, which everybody says is the largest component of the CAD industry and the one at which we are most at risk.

Twenty million dollars? Well, add up what we paid to acquire Cadetron in the first place, the money we spent subsequently bringing AutoSolid to market, the costs we incurred closing the office in Atlanta and moving development to Sausalito, the subsequent investment in AME/Eagle leading to its shipment with Release 11, and I suspect you'll come up with a figure about that size.

I can't be totally disinterested in the fate of AME since, by building the initial prototype in July of 1989, I played a rôle in the transformation of AutoSolid from a $5,000 stand-alone product targeted at mechanical designers into a $500 component of AutoCAD addressed to a much broader market. I initiated that project because I thought it was a way to rescue a project I thought was going nowhere by aligning it with the way Autodesk has always done business. Rather than addressing a small market with an expensive product (by the

standards of PC software), we could bring solid modeling within the reach of anybody who could afford the price of AutoCAD. My goal at the time was to "Within one year, sell more solid modeling systems that exist on the entire planet today."

Well, it took longer than I expected (everything does), but the Eagle group pulled it off, delivering, the day AutoCAD Release 11 shipped, a solid modeling extension that was far more comprehensive and ambitious than anything I had contemplated as an initial adjunct to AutoCAD.

And then. . . ? Silence.

Where was the large-scale, high-profile, roll-out of what could easily be adjudged the single most significant event in desktop design since 3D? Where were the advertisements and brochures that properly heralded it as a price/performance breakthrough comparable to the introduction of AutoCAD in 1982? Something like:

> "AutoCAD's OK, but what have you done for me lately?"
>
> How about solid modeling for $500?
>
> For years, designers have struggled to build complex models of three dimensional objects. Repeatedly, they have begged for relief from arcane commands, obscure terminology, and facilities that seem designed more to humble the design professional than to help him. "Why can't I have a system that works like the real world, one that lets me bore holes, mill, weld, and assemble pieces from parts with operations I can understand?"
>
> "Because that would take solid modeling!" was the answer. "That's a sophisticated technology, suitable only for high-end mechanical engineers, far too costly for your needs and requiring much more computer than you could ever afford."
>
> Until today. With the shipment of AutoCAD Release 11, Autodesk announces the Advanced Solid Modeling Extension, which delivers true, thoroughly professional solid modeling as an integral component of AutoCAD. And, in keeping with Autodesk's commitment to its customers, it runs on the same affordable machines that run AutoCAD, costs less than $500, and, through open architecture, encourages users to build application systems upon it.
>
> CAD before AutoCAD was an elite club, foreclosed to the vast majority of designers who couldn't afford expensive mainframe computers. Just like solid modeling before today. With AutoCAD Release 11 and ASME, we're putting an end to that, forever. Welcome to the golden age of engineering.

This was how we announced AME in the Release 11 press release of October 18th, 1990.

> AutoCAD Release 11 supports the optional Advanced Modeling Extension (AME) which gives designers and engineers powerful constructive solid geometry capabilities that are completely integrated within AutoCAD. With AME, designers can create complex, three-dimensional models by constructing them from simple 3-D shapes.

If this were any more low key, it would be apologetic.

The following sentence closed the paragraph on AME that appeared on the second page of the *Autodesk Designer*, a flyer mailed to Autodesk dealers, dated October 15th, 1990. AME appeared next to last in the list of Release 11 benefits, right before "Personalisation."

> AME's price, US$495, is unprecedented for solid modeling software, and is sure to introduce the benefits of solid modeling to a wider customer base, especially in the mechanical engineering market.

Well, gosh, I couldn't have put it better myself, but the very tone of this sentence, examined closer in the context of the rest of Autodesk's promotional material, speaks volumes about the assumptions that underlie Autodesk's do-nothing posture toward its products. Indulge me for a moment while I stick this sentence with a pin and pick it apart under the magnifier. "AME's price... is sure to introduce the benefits of solid modeling to a wider customer base...." Precisely how? What is the chain of cause and effect? How does the *price* act to *introduce* the *benefits*. The price, in other words *the mere event of Autodesk's making the product available*, is seen as an actor in the market, empowered somehow to set in motion the events Autodesk wishes to transpire.

This isn't putting the cart before the horse; it's expecting the cart to go with no horse at all. A low marginal price creates the *conditions* under which Autodesk possesses an *opportunity* to transform solid modeling from a highly specialised niche market into another widely-used application like AutoCAD and perhaps, by doing so, to break down the barrier that has kept most designers from truly entering the world of three-dimensional modeling, creating, in time, a market for additional design tools as large or larger than the current AutoCAD drafting market. All the work that went into AME from the inception of The Engineer Works at Cadetron in Atlanta through the breaking of the champagne bottle on the UPS truck the day Release 11 left Sausalito created only the *potential* for success, conditions that were necessary but not sufficient.

For all the wonderful things to happen which so many people worked to bring about, a few more links in the chain of causality need attending to. Users must learn of the existence of the product. Its benefits must be explained to them. They must understand both what it can do and its limitations. And they need the opportunity to evaluate it for themselves.

These are all the things we had to do between 1983 and 1985 to convert the potential of AutoCAD, the computer program, into the success of AutoCAD, the new world standard for CAD. Having achieved success once does not grant us a license to succeed with additional products, whether related to AutoCAD or not, without doing all the same things we did to bring AutoCAD before its potential customers.

Yet today, Autodesk attends fewer trade shows, garners less press, communicates less frequently and in fewer ways with its user community. What other software company comparable to Autodesk has no user newsletter? What other software company refuses to provide technical support to users in need?

There is a dangerous myth that because we have a reseller channel, we needn't do the things other companies must to create demand for our products. What nonsense. Pushing products into a distribution channel is like pushing on a rope. Distribution is an asset only if the product is pulled out the other end; if customers are brought to the reseller seeking the products for sale there. The responsibility for creating that demand rests primarily with the manufacturer; after all, it is he who keeps the majority of the money from the sale. Manufacturers who neglect this simple, eternal truth of retailing may, in the short term, post better profits but before long will suffer, along with their resellers, the symptoms of declining sales, falling earnings, and eroding market share.

How often do you see an advertisement from Autodesk in the publications you read? Compared to 1984 and 1985, how frequently do you see articles in the press about the myriad applications of Autodesk's products? Compare the visibility of AutoCAD, for example, to that of a typical Microsoft product such as Word or Excel. Immediately somebody shouts, "But those are mass-market products, not highly specialised products like AutoCAD. Besides, they're addressing a much more lucrative market." Well, let's see. Microsoft's sales are about five times ours. Of that, about half is application software, so all the Microsoft applications, including Word, Excel, PowerPoint, Project, and Works, add up to about 2.5 times our sales. If you assume Word and Excel account for the lion's share of this revenue, that means the sales of these products are roughly comparable to Autodesk's revenue from sales of AutoCAD. So in fact the larger volume of these products is just about

balanced by their lower retail price, yielding the same revenue. Word and Excel ads are everywhere. Where are the AutoCAD ads?

"You can't sell a product like AutoCAD the way you sell a spreadsheet. It's a different market, and it has to be addressed in a different way." This claim might be credible if, years ago, people hadn't insisted you couldn't sell *spreadsheets* the way Microsoft sells spreadsheets. Remember when spreadsheets were vertical market tools for financial analysts in the Fortune 500? It was only after the products were mass marketed, widely available, and affordable that the market for spreadsheets exploded, including today scientists, engineers, high school students, and diet book authors. It was this same kind of expansion of the market for CAD, set into motion by Autodesk's early and highly successful though meagerly-funded communication efforts, that redefined CAD as something suitable for "anybody who draws."

"But advertising is expensive! There are more cost effective ways to getting the job done." Surely. And advertising and other paid promotion should be but components of a balanced program including trade shows, co-promotions, dealer incentives, and all the myriad ways market-savvy companies stimulate demand. If Autodesk were achieving high visibility in these other ways, one might conclude that advertising was unnecessary. But we aren't. In fact, I believe Autodesk is increasingly slipping from sight, except within the existing community that uses its products. Talking to them is important, but it won't expand the market; we're preaching to the choir. To build markets you have to go out, make some mistakes, find what works, then build upon it. And that costs money. Once you realise that the revenue from a major Microsoft application is comparable to the sales of AutoCAD, the invisibility of AutoCAD is even more inexplicable since Microsoft's margins are the same as Autodesk's. Microsoft isn't doing all that aggressive marketing by spending more on a percentage basis. They're either doing less of the things that don't get them in front of the customers, or they're getting more for their money.

Advertising is, of course, the last resort of the communicator. Autodesk was able to promote AutoCAD in the early days with very little direct advertising by gaining editorial coverage in a wide variety of publications. A five or six page story about a user's success with AutoCAD delivers many times the impact of an advertisement at a fraction of the cost. These days, however, AutoCAD applications have become far more common and more imagination is needed to get the attention of the press. Imagination is something that's never been in short supply around Autodesk, yet we seem to consistently squander the visibility it gains us through lack of follow-through. One of the reasons I started the cyberspace project was to create a high-profile, exciting technology project to make the company stand out in the industry. Well, at least that part worked! Within a year, Autodesk was mentioned in the technology focus column and later on the front page of the Wall Street Journal, in the New York Times, and in many other extremely hard-to-crack publications in which paid advertising is forbiddingly expensive. And did we effectively communicate to any of these writers, given the entré created by the cyberspace project, the Autodesk story, of how this project indicated our ongoing commitment to lead the three dimensional design market from the cutting edge? Well, no we didn't. That story, and with it the equivalent of several million dollars of paid publicity simply slipped through our hands. Or consider the month when *Scientific American* ran a screen shot from one of our products on the table of contents page and devoted two pages to one of our new products. Did we use that opportunity, in the same publication where Autodesk ran its first ambitious four-colour advertisement, to showcase the company and its mainstream products? No, we were identified by the columnist as a "California computer games company." This would never have happened in 1984.

Foregone opportunities don't show up on the profit and loss statement, at least not right away, nor are they ever itemised and charged back to internal departments. But each one is the equivalent of burning current dollars and bypassing future opportunities.

If AutoCAD's invisibility is not in keeping with its importance, then the consistent lack of support for new products makes their failure inevitable. We spend large sums developing a product, ship it, ignore it, and it fails. After a while, nobody's interested in promoting new products because "they all fail." And eventually, so does the company. Ignore the subtler points of strategy and look only at the numbers. Autodesk is committed to increasing its sales and earnings at a rapid pace for the foreseeable future. The price/earnings premium on our stock reflects an assumption we will succeed in this. Since AutoCAD already commands a large share of the current CAD market, we cannot achieve this growth by taking business away from competitors. Consequently, the growth objectives can be met only by broadening the market for AutoCAD, thereby increasing its sales, or by launching new products which, in time, will contribute revenue and earnings comparable to AutoCAD. But if we don't promote AutoCAD, how is its market to grow? And if we push each successive new product off the loading dock, keening our ears for the thud that indicates "another Autodesk new product flop," how are these products to help us? The absence of effective promotion of either AutoCAD or our new products precludes success through either path.

"Sure, we'd like to do all those things, if only we could afford them, but the money just isn't there in the budget to do the kind of advertising, promotion, and public relations you're suggesting."

Out of two hundred million dollars a year?

The Risks Of Caution

> No lesson seems to be so deeply inculcated by experience of life as that you should never trust experts. If you believe doctors, nothing is wholesome; if you believe theologians, nothing is innocent; if you believe soldiers, nothing is safe.
>
> —Lord Salisbury, 19th century British prime minister

How has Autodesk come to such a sad state? Surely our management is not incompetent, nor bent on the destruction of our company. After all, they are for the most part the same people who led the company through the times of its greatest triumph, when Autodesk built the initial success of AutoCAD into one of the premier entrepreneurial success stories of the 1980s.

No, I think the problem that afflicts Autodesk, its seeming inability to act in any way whatsoever, stems from a failure of confidence, the self-assurance in the face of uncertainty that what we are doing is *right*, which is essential to any entrepreneur. When confidence ebbs, the courage to act dies with it. Successful businesses are rarely if ever built by following an obvious path, and the actions that created great industries often seem clear only in retrospect. The entrepreneur needs the right mix of brash disregard for the general consensus which usually tells him he is a fool or worse, tempered by firm grounding in the realities and potentialities of the arena he is operating in and the dynamics of the marketplace. These allow him to weigh his chances of success against what is at risk in the venture. An entrepreneurial company needs this balance as much as the founder of a start-up, and it must not only seek individuals who embody these properties, it must structure its organisation to achieve the right balance between imagination and caution.

It is an imbalance, or more accurately a disconnection, between these qualities which I believe lies at the heart of Autodesk's problems.

The stainless steel web

When self-confidence fails, caution rules. When timidity and unrestrainted risk-aversion gain the upper hand, the kind of imaginative and bold initiatives that companies must make in order to sustain their rapid growth are forced to run a gauntlet of analysis and criticism that no suggestion, no venture not already proven successful, has a chance of surviving.

This is the source of paralysis, why companies fail to act even when there is, within the company, a broad consensus that problems exist, action is needed, and even agreement on the nature of the changes required. Even then, each specific recommendation finds itself stuck in a stainless steel web; the chances of it surviving all the individual sign-offs needed to be implemented are remote.

Here are some strands of the stainless steel web.

- ❑ "It's too big a project for the next release."
- ❑ "The users aren't ready for it."
- ❑ "There's no way to know how big the market is."
- ❑ "The market today isn't big enough."
- ❑ "The product support burden would be too high."
- ❑ "That would be competing with the developers."
- ❑ "We can't change the ADI spec to accommodate it."
- ❑ "That won't work on the Apollo/VAX/Cuisinart."
- ❑ "It would complicate the documentation."
- ❑ "It would be too difficult to test."
- ❑ "Nobody's asked for it."
- ❑ "The dealers couldn't sell it."
- ❑ "It would increase profits, but reduce margins."
- ❑ "We'd have to change all the drivers."
- ❑ "We don't have any experience in that area."
- ❑ "It would add a disc to the release."
- ❑ "It would confuse Autodesk's image in the Fortune 500."
- ❑ "We'd have to renegotiate our contract with. . . ."
- ❑ "Another company tried it five years ago and failed."
- ❑ "That wouldn't work on pen plotters."
- ❑ "It would divert resources from. . . ."
- ❑ "We should wait for the database redesign/objectification/second coming."
- ❑ "It'll cannibalise sales from. . . ."
- ❑ "None of our competitors does that."
- ❑ "That will always remain a mainframe application."
- ❑ "It could lead to a channel conflict."
- ❑ "It would be too expensive to introduce."
- ❑ "That's platform-specific."
- ❑ "We'd have a hard time explaining it to the analysts."
- ❑ "If you give 'em that, next they'll want. . . ."
- ❑ "How will you do dragging/picking/drooling. . . ?"
- ❑ "There's no synergy with AutoCAD."
- ❑ "We should wait until there's a standard for. . . ."
- ❑ "The dealers will think we're going around them."

- ❑ "If we include it, we can't charge extra for it."

Each and every one of these concerns may be valid, and all of them, to the degree appropriate, should be weighed when considering a new product, changes to an existing product, or revisions in our pricing, discount structure, distribution channels, or marketing strategies. The difference between prudence and circumspection and becoming ensnared in the stainless steel web is the difference between a can-do spirit and a can't-do, passive outlook on the world, between action and reaction.

Paralysis in the face of opportunity sets in when the valid concerns and genuine difficulties posed by any initiative become viewed not as obstacles to be overcome, but insurmountable barriers blocking its implementation. If any proposal is subjected to an initiation ritual in which any black ball causes its rejection, the chances of any redirection in strategy occurring becomes vanishingly small. And as a company expands, diversifies, and operates in a larger and more complicated arena, the check-off boxes that can block action proliferate. As a suggestion is reviewed, its near-term difficulties and immediate risks come to outweigh the benefits which, though potentially large, are deferred into the future and subject to uncertainty.

In such an environment, an unambiguous statement of direction, strong and effective leadership, and continued follow-through by senior management is essential if the company is to progress. Otherwise, the parochial concerns of individual departments will block any and all changes to the way they've become accustomed to doing their work.

This is the stainless steel web—the timidity trap. It seizes mature companies who can always seem to amass a hundred reasons *not* to do something against one or two reasons in favour, even if the potential benefits include saving the company. It supplants the entrepreneur's approach, "try it and see if it works," with the central planner's: "let's do some market research and develop a matrix for evaluating this proposal." It is the abandonment of judgement in favour of calculation.

You cannot lead an industry by studying the actions of your competitors. To lead, you must understand the mission of your company and take the steps which, in time, will be studied by other, less successful companies seeking to emulate your success.

Challenges and *Challenger*

Recently, I have been approached by more people concerned for the future of Autodesk than at any time I can recall. I remember no prior occasion where there was virtually unanimous belief among the people I count as key contributors throughout the company—in no way limited to the technical staff—that Autodesk is on the wrong course and that the destruction of all that we've worked for is simply a matter of time.

Further, I have never before encountered the resignation, frustration, and despair I see today. Before, when Autodesk seemed about to do something wrong, people would raise their voices in protest and, if necessary, throw their bodies in front of the wheels to save what they believed in. Problems were seen as opportunities, spurring action to fix them. No longer. Many of the people who did so much to make Autodesk what it is today believe now that they are ignored, dismissed out of hand, scorned, or not trusted. And I think they are correct.

To me, this is more frightening than any of the details of shortcomings in our products or missed opportunities to promote them. It indicates a breakdown of the flow of information in the company, from bottom to top and from top to bottom, which can set in motion the events that lead to disaster.

This reminds me of nothing so much as the report on the space shuttle *Challenger* accident of 1986. NASA seemed to have been sleepwalking its way to that tragedy in an environment where engineers in the ranks knew of problems and groused over being ignored when they raised them, where management heard of problems but was never made aware of their significance or the potential consequences of inaction, and where the pressure of a tight schedule and attempts to meet unyielding demands with inadequate resources caused everybody involved to lose their perspective and make judgements they would have immediately realised were flawed, if only there had been enough time to *think* about them.

Little or none of the industry background, the description of problems in the company, the worries about the state of our products, or the recommendations for addressing these matters originated with me. I have departed from my usual policy of attributing everything to its author both because many of the worries and recommendations are widely shared, and because I've decided to take all the heat personally for putting these issues on the agenda. I'm not a stalking horse for anybody, and I don't want anybody to suffer just because I happen to agree with them.

Autodesk is proud of its open door policy, and counts on it to bring the information before senior management that they need to set the course for the company. Such a policy can work only as long as people believe they are listened to, and that decisions are being made on grounds that make sense for the long-term health of the company. Rightly or wrongly, there is a widely-held belief which I'm articulating because I share it, that management isn't hearing or doesn't believe what deeply worries people throughout the company, and isn't communicating to them the reasons for the course it is setting. This is how bad decisions are made.

And that can lead to disaster.

The Nightmare Scenario

It was a chilly, grey, drizzly April morning. Whenever the weather was like that, and it was *usually* like that near Seattle, the ascetic wunderkind of the software industry was filled with nervous energy. "Damn," he said, looking out the window, "what I need is *more money*."

He looked at the most recent ranking of top-ten software companies, noting with satisfaction not only the extent of his lead in the number one slot, but that it was widening, and gazed further down the list seeming, for a moment, almost like a lion scanning a herd of antelope from a distance, weighing vulnerability against size.

"Excel's got them on the run," he muttered, looking at the faltering growth and eroding margins of a former darling of the software industry. "Hrmpfff... databases. Wait'll they see our...." His eyes continued down the page.

"Now that's interesting," he said, turning his mind in a direction he hadn't thought much about before. "Two hundred million plus sales, 24% after tax, 60% market share, no mass distribution, and they don't even have a product on Windows."

By the time he left the office that evening, the outline of the plan was complete. Implementation began within days. Once he'd realised that this not only had the potential to become the next spreadsheet from a revenue standpoint, but could be the keystone of his larger strategy to move the windowing system and portable operating system into the workstation market, and was, in addition, the largest remaining software market his company hadn't plucked from its pioneer, he gave the project the highest priority.

First, he appointed a strong project manager who'd proven himself on another mainstream application. He

delegated total authority to this person to get the job done and gave him discretionary authority over a generous budget and the go-ahead to recruit a project team from anywhere in the company. That person chartered an in-depth study of the market, the users, the target company and its products, and the existing distribution channels. Quickly the plans were drawn.

While negotiations were underway to acquire those technological components of the product more expeditiously bought than implemented in-house, a core team of 75 developers was assembled. At the same time, quality assurance, marketing, sales, and other departments began to assemble the cadres they would need as the product came to require their attention. What with the rapid growth of the existing products even during a recession, resources were tight and many of the jobs were filled by new hires. The staffing up worried the financial planners, but when shown a demonstrated market of more than two hundred million dollars a year waiting to be grabbed, they signed off on all the requisitions.

By July, the central graphics engine had been acquired and was being extended from 2D into a 3D user interface. Some of the developers were particularly amused when they adapted several ideas for object pointing and orientation in 3D from presentations made at a conference by members of a research project at the target company. Down the hall, the surface and solid modeling code that had been acquired was being hammered into shape and prepared for linkage to the user interface.

Once the outline of the product was clear, another 25 people were added to the project. This group was dubbed the "compatibility team," and immediately set to work developing translators to read and write the databases of the target product, those of its principal competitors, plus the government-mandated interchange standard, awful as it was. Most of the other translators were already in-house, components of the spreadsheet, word processor, or presentation graphics products and needed only to be adapted to the engineering product.

When you do things on this scale, it's hard to keep what you're doing under wraps. But that needn't be a disadvantage. A prototype of the product was shown as part of an industry briefing on the company's strategy for the CAD market. The product was rolled out in Detroit, overlooking the heart of the US engineering and manufacturing plant. The analysts anticipated the outlines of the announcement but were stunned by the details.

- Windows-based.
- Includes 2D illustration, 3D wireframe, 3D NURBS surfaces, and CSG.
- Integrated photorealistic rendering.
- Integrated animation.
- DDE hard links to spreadsheet, word processing, and database servers.
- Reads and writes AutoCAD **`DWG`** and **`DXF`** formats.
- Built-in macro recorder and editor.
- On-line manual and tutorial.
- Integrated drawing management system.
- 1.2 gigabytes of clip-art, samples, and symbols on 2 CD-ROMs.
- "Applications Bridge" allows AutoCAD ADS applications to run with minor modifications.
- Retail price $895, AutoCAD users can trade up for $100.
- Available through all software vendors.

Company officials announced plans to introduce finite element analysis, numerical control machine programming, and other engineering modules over the next several years. "The personal computer has brought a new era of productivity to business, and now it's time to do the same for design, engineering, and manufacturing. Next January, we'll deliver a product that everybody can afford and anybody can learn to use; the first CAD

product to break down the barriers between the front office, the design studio, and the factory floor by sharing a common user interface, hardware and software environment, and file formats," said the founder, closing the presentation. As he rode down in the elevator he said privately to the project leader, "That should give 'em something to chew on in Sausalito, don't you think?"

Autodesk's stock fell 25% the next day, and although it recovered half the loss over the next several weeks, the attention of management was consumed for much of the next month in talking to analysts, institutional shareholders, dealers, developers, and key corporate clients. In addition, there were depositions to be taken and documents to be produced in conjunction with the two shareholder suits filed days after the announcement. Product development schedules were advanced, low-priority projects were terminated and staff shuffled around. A crash project was set into motion for a promotional campaign built around the "world standard" theme. But somehow it didn't seem to matter any more.

Windows Engineer was featured in a two minute spot in the Superbowl next January, the game where, ironically, San Francisco lost in the last minute. Shipments commenced the following Monday, the same day the six page insert ran in the Wall Street Journal and the ad campaign broke in the magazines. The database-compatible Macintosh version was introduced the following July.

Within a year, their market share exceeded 50%.

Marginalia

> Prosperity is an instrument to be used, not a deity to be worshiped.
> —Calvin Coolidge

Are we killing our company by making too much money? I believe this is a possibility, and I'd like to explain the situation and present some ways of remedying it. But first, since it involves some fairly arcane financial concepts, some background is required. I covered these matters in more detail in Information Letter 12, written on July 20th, 1985 (see page 280). Here I'll give only a brief summary, sufficient for understanding the discussion that follows.

When a company sells something and gets paid for it, that's *Sales*. Out of that, the company has to pay its bills: the cost of raw materials, salaries, rent, commissions to sales agents, interest on debt, and so forth. What's left is *Pre-tax Earnings*, and if it's less than zero the company is losing money. Next in line are the tax collectors, and what doesn't go into their pockets is *After-tax Earnings*, or plain *Earnings*. Dividing Earnings by Sales gives the *Margin*, the percentage of sales that ends up as profit. If the company is a stock corporation, you can divide the total earnings by the number of shares outstanding and get *Earnings Per Share* or *EPS*. When the stock is selling at a given *Price*, dividing the share price by the EPS gives the *Price/Earnings Ratio* or *P/E*.

These numbers indicate the general state of health of companies and industries. Here are some real numbers for real companies.

Company	Sales ($ millions)	Earnings ($ millions)	Margin	P/E
Autodesk	237.8	56.7	24%	22
Microsoft	1183.4	279.1	24%	34
Intel	3921.2	650.2	17%	14
Apple Computer	5558.4	474.9	9%	16
Lockheed	9958.0	335.0	3%	8
Motorola	10885.0	499.0	5%	15
Bank of America	11389.0	820.0	7%	9
Proctor&Gamble	24081.0	1602.0	7%	17
AT&T	37285.0	2735.0	7%	13
IBM	69018.0	6020.0	9%	11
Ford Motor	97650.0	860.1	1%	18

Amazing, isn't it, to think that Autodesk and Microsoft make more than *two and half times* the profit per dollar of sales than IBM? Isn't software neat? Looking at the P/E column indicates investors think it's *awfully* neat. So neat they're willing to bid up the price of a share of our stock or Microsoft's to between 22 and 34 *times* the yearly earnings that share represents, while they're only willing to pay between 8 and 18 times the earnings for companies with sales dozens of times larger and histories spanning decades.

The reason software stocks command these high premiums (which translate directly into the price of the stock), is that they've demonstrated they can run large profits while continuing to grow rapidly. This is usually the case for companies in a rapidly-expanding market. Once the market becomes mature, growth slows, market share battles erupt, and margins fall as companies spend more and more winning customers from their competitors.

Investors and analysts have learned to watch a company's margins closely. Changes in margin are often among the earliest signs of changes in the fortunes of a company, for good or for ill. When sales, earnings, and margins are rising all together, it usually means the market for the company's products is growing even faster than the company anticipated; the future seems bright. When margins begin to decline, however, it can indicate the company has let spending outpace sales. When competition begins to affect the company, or even when a company fears future competition, it may spend more on promotion, accelerate product development, and offer incentives to dealers and retail customers—all reflected in falling margins.

But high margins aren't necessarily a good thing, particularly in the long term. One way to post high margins is by neglecting investment in the company's future. Any profitable company can increase its earnings and margin in the short run by curtailing development of new products and improvements to existing products, by slashing marketing and promotional expenses, and by scaling back the infrastructure that supports further growth. Since there's a pipeline anywhere from six months to several years between current spending and visible effects in the market, sales aren't affected right away. So, with sales constant or rising slowly and expenses down, earnings and margin soar and everybody is happy.

For a while, anyway. Eventually momentum runs out and it's obvious the company can't sustain its growth without new products, adequate promotion, and all the other things that constitute investment in the future of the business. It's at that point the company becomes vulnerable to competitors who took a longer view of the market.

One of the most difficult and important decisions the management of a company makes is choosing the level of investment in the future of the business. Spend too little, and you're a hero in the short term but your company doesn't last long. Spend too much, and the company and its stock falls from favour because it can't match the

earnings of comparable companies. Unlike many choices in which there is relatively little room for maneuver, the level of reinvestment can vary widely. After all, Autodesk could have spent $40 million more last year and still earned more on every sales dollar than IBM.

When a company is running margins too high to sustain, the situation can be discerned by the following kind of symptoms. Product release dates are stretched out, and each product release contains less substantive content. Marketing and other promotional activities are cut back, abandoning products to sell themselves. Budgets for the development and promotion of new products are slashed, sacrificing future sales and earnings from those products to current earnings.

Sound familiar?

It's how a company dies by making too much money.

Money in the bank

Let's turn now to what happens to the money that remains after all the bills and taxes have been paid. A small amount is paid back to the shareholders as dividends, but the overwhelming percentage goes into the corporate treasury—the bank account—the money bin. When a company runs the kind of margins Autodesk does for all the years we have, that adds up to a tidy sum: in Autodesk's case more than $140 million. When thinking about the future of the company, what can and can't be done with that cash is vital to understand.[341]

At the simplest level, the money belongs to the company and management can do anything it wishes within the law: give some back to the stockholders as a special dividend (as we did in 1989), buy other companies (as in the Generic Software acquisition), buy real estate or other capital goods for the company (for example, the scheme to build a "campus" among the cows), or just invest the money, collect the income, and add it to earnings. The overwhelming percentage of Autodesk's cash is currently invested in safe, short-term interest bearing securities, which is why I sometimes refer to Autodesk as a "combined high-tech company and money fund."

But here's the essential point. When you spend a dollar, whether to hire a programmer, buy a truck, run an ad, or take over Chrysler, it *it doesn't matter whether it came from the bank account or from current sales*. Many people think that because Autodesk has $140 million squirreled away (not counting the ball of string or the First Dime), we can use that money free of constraints. If I ran the zoo, I'd change the accounting system to cut some slack for companies that put away profits against future needs, then dipped into the cookie jar when an opportunity presented itself or an unanticipated risk emerged. But that isn't how it works. Regardless of how prudent you've been piling up money over the years, the moment you spend any of it in your business, it's just as if you increased your day to day operating budget. That means rising expenses without an increase in sales, and that translates into... falling margins.

About the only thing you can do with the money that *doesn't* cause margins to fall, other than giving it back in dividends, is investing it in other companies. When you make an investment, that's carried on the books as capital. As long as you don't have to write the investment off, it doesn't affect your operating results. (Outright purchases of companies or large percentage investments force you to merge or *consolidate* their earnings with yours, however, creating the risk of falling margins.)

[341] The discussion that follows is a much-abbreviated and simplified of the in-depth treatment of this issue in "The New Technological Corporation" on page 475.

The accounting for money in the bank, then, can create a situation where pressing company needs remain unmet because the expenditures required would cause margins to fall, yet at the same time, the company is actively investing its cash hoard *outside the company*, in other businesses, because those investments do not show up as current operating expenses. Thus, the accumulated earnings of a company, the ultimate result of its success, can benefit any venture *except* the one that made the money in the first place.

Sound familiar?

Margin vs. management

When it's new to you, some of the ways accounting affects companies such as Autodesk can be difficult to comprehend. However, it's essential for understanding the situation we're in.

Management strives, quarter by quarter, to meet the sales and earnings expectations of the Wall Street analysts and to avoid erosion in the margin which would be seen (rightly) as an early warning, presaging problems in the company. In the absence of other priorities this is foremost, as the consequences of a stumble can be dire. Remember when Autodesk's stock lost 25% of its value in a single day last January not because the company lost money or because sales fell, but simply because they didn't go up as expected? Just imagine what it's like when things get really bad. Or better, don't try to imagine; you probably can't.

But management has a more serious responsibility to the shareholders; to provide for the future of the company and its products. Focusing exclusively on this quarter's or this year's margins to the extent that industry averages dictate departmental budgets for our company is confusing the scoreboard with the game. We've seen how management strives to deliver numbers each quarter that fall close to the expected results. But management is *required* to protect the future of the company. When these two priorities conflict, it is time for management to make those difficult choices they frequently speak of, and do what is best for the company, not to behave as if their freedom of action were constrained by numbers cranked out by somebody in an office overlooking Wall Street.

Again, it comes down to the need for management to *act* when necessary, using the enormous resources at their command, summoning the courage to take the heat, if necessary, for adverse short term consequences that serve the longer term interest of the company.

The incredible shrinking budget

To illustrate the difference between doing what's best for a business and doing what's best for the numbers, I'd like to tell a true story that happened not long ago at Autodesk. I'm singling out this case because it demonstrates both how accounting can affect decision making as well as the difference between treating the numbers as a master versus simply as a reflection of the underlying reality.

I attended a meeting in early 1989, where I heard a discussion of how, over the coming year, it would be necessary for Autodesk to reduce its sales and marketing budget to lower and lower levels. Walking in from the outside, I found this more than a little puzzling. After all, weren't we in the midst of a still-unbroken series of sales and earnings records? Wasn't this year expected to be the best ever? Weren't we finally achieving substantial sales of AutoCAD to the large companies and government?

True, but there was this little matter of accounting, you see. From time immemorial, most copies of AutoCAD

have been sold by dealers. To simplify the numbers, assume the retail price of AutoCAD is $1000, the dealer pays $500 for it, and all sales by dealers are at the full list price. So, for every copy of AutoCAD that ends up in a customer's hands, Autodesk gets $500 and the dealer gets $500. Autodesk reports the $500 as Sales, deducts expenses, pays taxes, and ends up with earnings, say $125, corresponding to a margin of 25%.

But suppose, instead, we sell the copy of AutoCAD to a Fortune 500 account: Spaceley Sprockets, perhaps? In that case, the numbers look like this (again simplified for clarity). Autodesk ships the copy of AutoCAD directly to the customer and invoices Spaceley Sprockets for the full list price, $1000. However, the sale was not made directly by Autodesk; the order was taken by one of our major account representatives, the equivalent of dealers for large accounts. When we get the check, we pay a commission to this representative. Assume the commission is $500.

Regardless of who bought the copy of AutoCAD, the financial result, the fabled "bottom line," is the same. There's one fewer copy of AutoCAD on our shelf, and one more installed on a customer's premises. Autodesk receives $500, and our dealer or representative gets $500. But oh what a difference it makes in the accounting! In the first case, where Autodesk sold the copy of AutoCAD to the dealer, that was the whole transaction; whatever happened to the copy of AutoCAD after the dealer paid for it has no effect on Autodesk's books. Autodesk sells, dealer pays, end of story. But in the second case, when Autodesk sells to Spaceley Sprockets, that appears on Autodesk's ledger as a sale of AutoCAD for $1000. The instant the $1000 shows up, however, we immediately cut a check for the commission, $500, and mail it to the representative, leaving the same $500 we'd get from the dealer. Same difference, right?

Not if you're an accountant! In the first case, Autodesk made a sale for $500 and ended up, after expenses and taxes, with $125, and therefore is operating with a 25% margin (125/500). In the Spaceley sale, however, the books show we sold the product for $1000, yet wound up only with the same $125. So now our margins are a mere 12.5% (125/1000). And if we only kept $125 out of the $1000 sale, why that must mean our expenses were $1000 - 125 = 875$ dollars! Of that $875, $375 represent the same expenses as in the dealer sale, and the extra $500 is the representative's commission which, under the rules of accounting, goes under "Cost of sales."

Or, in other words, comes out of Autodesk's marketing and sales budget.

That's why the marketing budget had to be cut. *To the very extent the major account program succeeded, it would bankrupt the department that was promoting it.* If we were wildly successful in selling AutoCAD into the big companies, Autodesk would make more sales, earn more profits, then be forced to cancel marketing program after marketing program as the price of success! All because the rules of accounting would otherwise show falling margins or a rising percentage of revenue spent on "cost of sales."

The purpose of this discussion is not to complain about the rules of accounting. You have to keep score somehow, and while one can quibble about this or that detail, as long as the rules are applied the same from company to company and people don't lose sight of what's really going on, there's nothing wrong with the way these different transactions are recorded. After all, business is supposed to be about making money, and after all the adding and subtracting, Autodesk made the same $125 from both sales.

Instead, what disturbed me so much about this incident was the way management seemed to be taking their marching orders from the accounting rules rather than the real world. Budgets were actually being prepared on the assumption that marketing and sales efforts would have to be curtailed to offset the increased "cost of sales" from the major account sales anticipated over the year. Think about it: here we were planning for what was anticipated to be and eventually became the *best year in Autodesk's history*, and yet were forced to cut our marketing and sales *as a direct consequence of its very success*. Carried to the absurd, if the major account

program astounded us and began to dwarf dealer sales, we would have to lay off the entire marketing and sales department to meet the budget.

This bothered me.

It seemed to me that what was called for here was not a plan for the orderly dismantling of marketing and sales department, but rather some effective outreach marketing of the company's own plans to the shareholders and analysts. Rather than sitting in our offices and *assuming* that changes in the numbers would be interpreted in a certain way, what was needed was to contact the analysts and major shareholders and, if necessary, go visit them in person and explain precisely what was going on. In essence, we'd be saying something like this. "Look, for years you've been hoping we'd crack the Fortune 500 in a big way, and we think we're going to pull it off this year. But because of the crazy laws in the US about pricing, we have to structure the transaction like this, and so every time we make a new sale, it looks as if our marketing budget went up when in fact nothing of the sort occurred. We know you're concerned about eroding margins, but we can't afford to pass by the Fortune 500 because of an accounting rule, and neither can we let our success there kill off our existing marketing and sales efforts. Here are our forecasts with and without the Fortune 500 program; you'll see how much it will really contribute if it works. To make things clearer during this period, every quarter we'll break out the effects of this accounting rule in our financial reports so you can see we haven't really let our cost of sales get out of control."

There are two different ways management can react to a problem; either passively: seeing their freedom of action constrained on all sides by history and expectations, or actively: looking at how best to deploy the enormous assets of the company to do what must be done. This incident illustrates how much outlook can affect outcome.

I don't know what ultimately happened in this case. I was so appalled by what I heard that I vowed to never attend another management meeting, and I never have.

Business Development

> A wise man will make more opportunities than he finds.
>
> —Francis Bacon, "Of Ceremonies and Respects," *Essays*, 1625.

As we've seen, if we use any of Autodesk's cash hoard directly in the business, it increases the expenses we report, reduces our margins, and makes the company look less profitable in the short term. If we just sit on the money, after a while investors quite reasonably become restive. "If you can't think of anything to do with the money, why don't you just give it back to us," they say, even though paying out profits as dividends subjects them (in the United States) to a second round of taxation.

The only way to use the money without directly increasing expenses is by investing it. These investments can consist of buying other companies outright, making minority investments in ventures, or purchasing products or technology which Autodesk can incorporate into its product line. Autodesk has, over the last several years, made investments in all of these categories.

Once it became obvious that Autodesk's growth expectations could be met only by finding additional products that generate revenue comparable to AutoCAD, using our strong cash position to acquire those products outside the company became a strong priority. This is as it should be; after all, Autodesk has no monopoly on good

ideas for products nor on people able to turn ideas into reality. However, we must look closely at the kinds of products being sought and the criteria being used in the search to decide whether the candidates for investment we'll identify have a chance of succeeding in the software market of the 1990s.

Unfortunately, unless the goals and priorities of Autodesk's current Business Development effort have been seriously miscommunicated, it seems to me embarked on a quixotic search for something which in all probability does not exist: "The Next AutoCAD." Autodesk needs additional products that contribute results comparable to AutoCAD, but to expect them to share the price, customer profile, and distribution of AutoCAD excludes any product within the mainstream of currently successful software packages.

We are said to be seeking to increase our sales by hundreds of millions of dollars a year by finding products which "sell for $1000 or more per copy and can be sold through our AutoCAD dealer network." So, in other words, we're betting the future growth of our company on our ability to consistently identify products which *sell for more than any other widely-distributed software* and will be sold exclusively by a distribution channel which has demonstrated itself *incapable of selling anything other than AutoCAD.*

What's wrong with this picture?

When you adopt unrealistic selection criteria, you find unattractive alternatives. The desiderata that Autodesk is seeking in the products on which the company's future will be bet would have excluded every single successful product introduced since 1982 by Microsoft, Lotus, Ashton-Tate, Word Perfect, and Borland. What are the odds Autodesk will find not one, but several products that these companies have missed?

You can always find an investment that meets your criteria, but if your criteria are out of whack with reality, you might as well blow your money at the track where at least you get to smell the horses. A literal search for "The Next AutoCAD" always ends up with dorky stuff like overpriced high-end project management software. What a concept: jumping into the very top end of a market where the entry level is dominated by Microsoft Project, then slugging it out with a company five times our size, selling a product at more than twice the price, through a distribution channel a fraction as large. Why we could blow them away just like Boeing Calc (I'm not making this up) obliterated Lotus 1-2-3! Yeah, sure. And I am Marie of Roumania.[342]

I'm all for business development; just look at how many products and investments I've brought in the door. But when we seek "The Next AutoCAD," as we must, we have to use a little more imagination. When we find "The Next AutoCAD" it will look just like the *last* AutoCAD did back in 1982—a non-obvious product in a market waiting to be created, with a large body of potential users who haven't ever really thought about how useful such a product might be. It's that kind of product, whether it sells for $50 or $5000, whether it's sold in bookstores, by dealers, or door-to-door, that promises the kind of exponential growth we seek. Most companies never find a single product that grows the way AutoCAD has. If we hope to find a second, then a third, and then more, we're going to have to look for products that play as large a rôle in defining the PC software industry in the 1990s as AutoCAD did in the eighties. Those products won't look like AutoCAD at all; they'll seem, at first glance, just as unlikely to have a future as AutoCAD did when we started working on it. But seen through the right kind of eyes, in the context of where we're leading the industry, they can be chosen with confidence because they're going our way.

[342] Everybody asks about this. It's the last line of one of my favourite Dorothy Parker poems:

> Oh, life is a glorious cycle of song, A medley of extemporania; And love is a thing that can never go wrong; And I am Marie of Roumania.

Winning

> A great part of courage is the courage of having done the thing before.
> —Ralph Waldo Emerson, *The Conduct of Life*, 1860.

The paradox of Autodesk's situation is this. Despite having spent more than thirty pages describing deep-seated and serious problems with our company, I continue to believe that no company in our industry possesses the opportunity that Autodesk does today. Autodesk can, by seizing the moment, changing as required, and resuming an active rôle in the industry, build its success in the CAD industry into a much broader position in the software market and, in the process, resume the rapid growth of the early years which some believe is impossible in a "large, mature company."

For this to happen, Autodesk must step back from day to day events and look at where the industry has come, where it is today, and where it is going. In this paper I've presented my views; whether you agree entirely, dispute this point or that, or consider my entire world-view wrongheaded, you'll probably concur, after reflection, that it's highly unlikely an industry that has changed so rapidly in the first 45 years of its existence will conveniently cease to evolve at the very moment Autodesk becomes set in its ways.

Some people may welcome change, but most of us hate it. We prefer a quiet, normal life, doing the things we have become accustomed to. But sometimes change is necessary. Almost 10 years ago, I started to think about what was wrong with Marinchip, the company I started in 1977. Perhaps, I thought, it was time for a new company, a new way of doing things. What was wrong with Marinchip was that it got stuck in 1978, but the times, and the game had changed—by 1982, the PC market was becoming something very different. It was time to do something new. It was painful to change—painful to even think about raising money, working with lots more people, dealing with distributors, etc., but it had to be done. It was done, and Autodesk is the result.

Now it's time to consider whether Autodesk has become stuck in the past; whether it's time for Autodesk to change and how. The changes may be unpleasant to contemplate and difficult to carry out, but they may be just as necessary and, if successfully made, as rewarding. As I hope I've convinced you through brutal repetition if not by artful rhetoric, if and when Autodesk wishes to change course, it has the resources at hand to accomplish anything we wish.

With the imagination and resolve to use those resources, how can the future be anything but bright? AutoCAD owns the market for the creation of geometric models and preparation of drawings from them. We have just placed solid modeling in the hands of hundreds of thousands of people for the first time, and we're poised to follow that up with even more powerful surface and solid capabilities. With Generic CADD and AutoSketch, we cover the CAD market from the bottom to the top, on all popular computers.

We sell, with AutoShade, the most comprehensive tool for producing lifelike images from the models we build, a rendering engine created by the acknowledged world leaders in the field. We own a majority interest in Xanadu, a hypertext system conceived by the person who invented the concept and coined the word for it, and developed by a group of people who have devoted years to designing a knowledge storage system for the ages that is as much of an improvement over anything in existence today as books were over oral tradition. And along the road to Xanadu, they have developed what may be the most comprehensive and innovative object oriented development system ever built.

We own a majority interest in a company that has developed the first totally automatic, computer mediated information market. If ours is, indeed, the "information age," then an information exchange will be as funda-

mental a part of that age as a stock exchange in the age of capitalism... and Autodesk owns the only one in existence. We have acquired exclusive marketing rights to the only molecular design software that spans the price-performance spectrum from laptops to parallel supercomputers; positioning us in another market at the very beginning of its rapid growth.

We market the very best paint package for the IBM PC, which also happens to do animation. With a few technical tweaks and a major market repositioning, it and its successors could own the entire raster graphics segment on the most widely used computers. We sell the first totally integrated 3D modeling, rendering, and animation software for mass market machines. If sold at a mass market price, it could begin to build this new market just as AutoCAD did years ago. We have $140 million in the bank, a thousand of the best people in the world, presence in every major market around the globe, and a channel of distribution dedicated to selling our products.

These are resources as great, or greater, than Microsoft has used to launch their last *five* successful products. What is lacking? How can a company with such assets have problems? What is needed is simply a clear statement of where to go, and a commitment to get there.

How to take over the whole graphics industry in a year or so

> A good plan, executed now, is better than a perfect plan executed next week.
> —George S. Patton

Here is the kind of strategy I think a reinvigorated, aggressive, Autodesk should be unleashing on the industry. I have plotted this strategy based on what I know Autodesk *can* accomplish, not constrained by what Autodesk has convinced itself is "realistic." My plan combines short-term, remedial steps aimed at reducing risks created by our slow progress in recent years, repositioning of existing products to better fit today's market, and development of new products to close gaps which endanger our competitive position. These recommendations are a starting point, not a complete prescription for turning Autodesk around; they are intended to highlight the extent that Autodesk truly controls its destiny and to suggest other similar moves to consolidate and expand our market position.

None of these proposals would survive scrutiny based on the first third of the criteria I call the Stainless Steel Web; implementing any of them (or indeed, doing anything at all) requires, as a prerequisite, the will to act. And that is the hardest part of the whole scenario; the rest are all things we know how to do and have done before.

Manage our products. If we are to redirect our products and treat the market as something we control rather than react to, we will need not only direction from senior management but close day to day line management at the product level. Our products are too important not to have a single individual who can speak for them and promote them both within the company and without.

☞ **`PM1`**: **Appoint a strong product manager for every product.** Appoint or recruit a product manager for AutoCAD, and for every other product in Autodesk's line. That person should report to the President, and will be given essentially complete discretion over the development, marketing, and sales resources for the product, constrained only by the budget allocated to that product.

☞ **PM2**: **Implement product-level profit and loss accounting.** Structure our internal accounting so it becomes possible to monitor the costs incurred and revenue generated by product, allowing us to track the contribution of each product to our overall goals.

☞ **PM3**: **Integrate development and promotion budgeting.** Undertake no new product development efforts without a comprehensive plan, adopted in advance, covering both the development phase, the marketing roll-out, and the first two years of post-launch development and marketing support. Autodesk must never again abandon a product developed at great cost; doing so squanders the morale of developers as well as the company's cash.

By "product manager" I mean an individual with line profit and loss responsibility for the product; a person delegated the authority to decide what goes in a product and what does not, when releases are to be made and what they will contain, and who bears the responsibility for achieving the schedules associated with that product. The current function called "product management" would remain within whatever department it best fit.

Help the dealers. I'm concerned about the health of our dealers' businesses, and about the degree to which Autodesk relies upon them as our only channel of distribution. Still, I continue to believe that we should do everything we can to strengthen our dealer channel, and should abandon it only if its collapse imperiled Autodesk's survival. Autodesk and our dealers depend on one another, yet the nature of our businesses are very different. When our dealers are in trouble, it is incumbent upon us to look beyond the obvious in seeking ways to help them. Here are two examples of ways Autodesk could help our dealers survive.

☞ **DE1**: **Provide dealer inventory on consignment.** Autodesk is cash-rich; our dealers are starved for capital. Yet Autodesk requires its dealers to tie up a substantial amount of their working capital keeping copies of AutoCAD in stock. In addition, dealers must carefully estimate inventory levels to ensure they are able to deliver our products without waiting to receive new stock from Autodesk. We could drastically reduce the capital our dealers tie up in inventory by moving to a consignment stocking policy. Any dealer in good standing, not on credit hold, would be able to stock a quantity of AutoCAD equal to his average monthly sales without paying Autodesk in advance. As each copy was sold he would report its serial number to Autodesk and a Net 30 invoice would be generated for it. As long as the dealer collected from his customer before the invoice came due, he would have no capital tied up in flooring inventory of AutoCAD.

Spare me a recitation of the ten or fifteen reasons we could never adopt such a policy. Yes, you'd have to explain it to the analysts, live with greater uncertainty, and trust the serial number tracking mechanism to prevent cheating. But isn't this how a company with $140 million in the bank should behave when its cash-poor dealers are facing extinction?

☞ **DE2**: **Consider reducing the price of AutoCAD.** Pressure from dealers is often cited to justify increases in the retail price of AutoCAD. But it may well be that *lowering* AutoCAD's price is the best way to help our dealers survive. Autodesk has, over the last several releases of AutoCAD, continued to raise the suggested retail price and has therefore increased the price dealers pay for AutoCAD since this is a constant fraction of the retail price. Evidence suggests, however, that the actual price customers pay when they buy AutoCAD from dealers has increased, if at all, much more slowly than Autodesk's posted list price. This has resulted in a squeeze, where the AutoCAD dealer makes less and less on each copy of AutoCAD he sells, while being forced to bear ever-increasing costs to maintain an inventory of the product on the shelf.

For years, many successful AutoCAD dealers have said that the bulk of their profits come primarily from sales of computers and peripherals associated with AutoCAD, not the software itself. If this is the case, a primary

reason our dealers are encountering difficulties now is that the margins on their hardware sales are falling, squeezing the segment of their business which accounted for most of their earnings.

If we were to reduce the price of AutoCAD and that resulted in increased retail volume, would that not benefit the dealers? If the majority of our dealers derive more revenue from hardware and services sold with AutoCAD than from the software, increasing the unit volume of AutoCAD would actually result in higher sales and earnings, since increased hardware sales would more than compensate for the lost revenue from a lower price for AutoCAD. There's no question that cutting the AutoCAD price would result in a short-term hit to Autodesk's sales and earnings. But what is the net present value of preserving the channel which has accounted for essentially all our domestic sales of AutoCAD since 1982?

This is a case where the "safe decision" is to do nothing, but where inaction may accelerate the collapse of our dealers. There are alternatives to a simple price cut; we could create a "2D-only AutoCAD" and price it at, say, $1895. Or, we could introduce the 286 version of Release 11 at a lower price, for example, $2195 (this would require a modification of recommendation **AC1** below—I trust you can see the obvious way to reconcile these goals).

Low cost marketing initiatives. In the early days of Autodesk, lacking any marketing budget whatsoever, we found a way to gain influence without spending anything more then the cost of goods in our products. Sacrificing such opportunities now in a quest for illusory short term sales is folly, especially when marketing budgets continue to be tight.

☞ **MS1**: **Free product for "opinion leaders."** A small number of individuals exert great influence in our industry. These columnists, educators, editors, consultants, and analysts form the opinions which echo throughout the market. In the early days of building the market for AutoCAD, we sought to make each and every one of these individuals familiar with AutoCAD, even if it required retaining them as consultants, flying them to Sausalito, and asking them for their view of the industry following, of course, a half-day indoctrination in our product line. But usually, a free "evaluation" copy of AutoCAD was all it took to get their attention.

Now, I discover, Autodesk has become loath to provide free copies of our new products to influential people in the industry. I presume this is a misdirected attempt to boost sales instead of an urge to self-destruction, but the results are indistinguishable. When the Education department was created at Autodesk, Mike Ford gave it this charter, "Extract as much money as you can, but never lose a sale." Seeking revenue in the short term ignores the multiplier effect of having a class taught using your product, having a book written about it, or having it be the object around which a curriculum is developed. These benefits follow only when those who influence opinion get an opportunity to play with a product, and they're unlikely to buy every product that comes along to see which is worth their time. Providing free products to opinion leaders is a way in which software companies can create market presence comparable to the list price of their products, while incurring a cost equal only to the raw materials.

Near-term upgrade of AutoCAD. A $4000 list price product shouldn't look stingy. We should take whatever immediate steps are possible to make the purchaser of AutoCAD feel he is being treated as the elite customer he is; that he is dealing with a vendor who values the relationship with him.

☞ **AC1**: **Universal 80x86 release.** Include all operating system platform versions in the Intel architecture release. In other words, when you buy an 80x86 AutoCAD Release 11 you get 386 DOS, 286 DOS, Xenix, OS/2, and Windows executables all in the box. If any of those versions aren't available at the time we ship the

product, include a coupon for ordering the others (just like the coupon Microsoft includes for no-cost ordering of 360K floppy or CD-ROM versions of their products). When you consider the costs of separate inventories, bills of materials, exchange order processing, etc., the cost of this to Autodesk may be *negative*.

☞ **AC2**: **Eliminate AME surcharge.** Include AME at the $3500 price point. We want to create a huge new market for solid modeling, and we have a unique opportunity to do so. Coming down the pike is ACIS-based AME-2, a major functionality upgrade. What better way to sell it than to get every Release 11 customer acquainted with what solid modeling can do in his application area? This also eliminates the expense of processing AME A-codes.

☞ **AC3**: **Promote solid modeling.** It isn't too late to beat the bass drum about what we've done for the industry by including solid modeling in AutoCAD Release 11. Even if we continue to charge extra for AME (especially, in fact), we should give it the launch it deserved when it was introduced, even though several months have passed.

☞ **AC4**: **Expedite Windows AutoCAD.** The existing Windows AutoCAD Product Plan Proposal should be implemented in its entirety as the highest priority project associated with AutoCAD Release 11. If diverting resources from other platforms or related projects can help ship this product sooner or make it more comprehensively integrated with Windows, divert them.[343]

☞ **AC5**: **Expedite Windows-32 AutoCAD.** Assign resources appropriate to ensure that a 32 bit version of Windows AutoCAD will be available for delivery the day Microsoft ships the required version of Windows to customers. If more than six months can be saved and substantial performance improvements had by pursuing an interim 32 bit version (e.g., Watcom), consider it seriously.[344]

☞ **AC6**: **Include CD-ROM(s) with AutoCAD.** Autodesk has everything in its possession today to include a compact disc with every copy of AutoCAD we ship. Six hundred megabytes... what could we fill it with? To start with, all the stuff on the existing drawing library disc, which needs only to be run off in sufficient quantity and included in the package. How long can that take? As soon as we can arrange it, we should make space on the CD available to all our developers to include sample applications and demos of their products. Within 45 to 60 days, we should be able to include a machine-readable version of the AutoCAD manual, hot-link indexed and connected to the HELP command in the product. If everything won't fit on one CD, then two or three. We'll have to get rid of that overpriced retrieval program on the current CD. There's no need to replace it; simply providing a High Sierra format CD and a directory map is enough to get the files off on any platform I'm aware of.

☞ **AC7**: **Upgrade AutoCAD documentation.** Re-design and re-print the AutoCAD Reference manual and Tutorial to equal the production values of the corresponding manuals included with Microsoft Word or Corel Draw. This is entirely a question of graphical design, use of colour highlights, etc. In the longer run, develop an "AutoCAD Encyclopedia" to supplement the existing reference manual, organised along the lines of the Microsoft Word or Excel reference manuals.

☞ **AC8**: **Direct support line.** I think it's shameful to ask $4000 for a product and then refuse to talk to any customer who calls with a problem. We should provide, at the minimum, one year of free telephone consultation. There's nothing wrong with asking who his dealer is and providing feedback to the dealer, but we should take the call and ungrudgingly provide whatever assistance we can.

[343] AutoCAD Release 11 Extension for Windows was announced in March 1992 (see page 781).

[344] This was accomplished (using Watcom) in the shipment of Release 12 for Windows in February 1993.

☞ **AC9**: **User newsletter.** We should immediately start production of a quarterly AutoCAD user newsletter, to be mailed to all registered users. The content of the newsletter would primarily be meat and potatoes technical and application information about the product, questions and answers, and the like. For a prototype, look at the Cobb Group newsletters such as "The Expert" covering Excel. New product information, listing of interesting items available on CompuServe, and the like could serve as filler, but would not be the main purpose of the newsletter.

AutoCAD future development. Beyond the short-term actions to upgrade the quality of AutoCAD and its perception as a premium product, we must revisit our development plans for Releases 12 and 13. Does a release defined by a shipment date rather than customer requirements, one motivated by a hope that it will create short-term demand for the product and raise its "street price," keep faith with the customers who are responsible for our prosperity? How long can we get away with releases with less and less substance in each? What is an appropriate manpower commitment to development and maintenance of our principal product?

☞ **AD1**: **Developer product orientation.** Make the time and resources available, even if it means slipping other schedules, for every individual involved in the development of AutoCAD to become familiar, in depth, with a modern Windows or Macintosh application such as Excel, Word, or CorelDRAW! This should be viewed as an education in the standards of today's market and what users expect of products, not aimed at mastering the applications themselves.

☞ **AD2**: **Evaluate development and QA staffing.** Investigate the tradeoffs between manpower allocation, content of each release, and time to market, then propose staffing plans for various points on the curve. This investigation must include a review of development methodology and tools which permit more developers to work together without conflict.

☞ **AD3**: **Parallel Release 13 projects.** Take all of the contending plans for Release 13, set them up in parallel, and let them run for 6 to 9 months with the goal of producing, at the end of that time, a base suitable for all the requirements anticipated for AutoCAD over the next decade. These projects would, collectively, compete with the **NC1** project (see below), to become the base for future development. At the end of the parallel phase, if it isn't already obvious, an in-depth review will be made to decide which alternative(s) to continue to pursue.

Own the raster market. Multimedia's going nowhere. Why? It's too damned *expensive*. When you're trying to pioneer a new market, you have to recruit a small cadre of "early adopters" who become the fanatics who discover new applications for the product, build upon it, and generate, through their efforts, the first success stories for the product which are echoed by a well-designed promotional campaign. This is how we got AutoCAD going. If we had introduced AutoCAD at $3000, I am convinced that very few if any of the people who ended up so influential in its early growth (many of whom now work here—ask them), would have been able to afford or have been inclined to take a chance on it.

The multimedia market closely parallels the early days of AutoCAD; both were considered outside the mainstream of PC applications, both were limited at the start by crude and expensive hardware, and both lacked the collateral support resources (books, training, user groups) that allow a small initial beachhead to expand to a much larger community of users. So what do we do? Raise the price on Animator. Wizard.

☞ **RM1**: **Re-price Animator at $149.95, add graphics modes, target paint programs.** If we want to *own* this market, and I cannot conceive of any reason we wouldn't want to, here's how to go about it. First, we

have to reprice and reposition Animator. It's a spiffy animation program, but it also happens to be the best *paint* program available for the PC as well. Its only limitation is that the only graphics mode it supports is 320×200 VGA/MCGA mode. What I'd do is remove that restriction, and support all the 16 colour modes and SVGA 256 colour modes *for single-image painting only* (the easiest way to accomplish this is to license and use MetaWINDOW, as we used for *Chaos*). Then, we re-price the product at $149.95 and re-launch it, aiming it directly at Z-Soft's various Paintbrush products and Colorix VGA Paint. "For the price of a paintbrush, we give you the whole movie studio. . . ." That kind of thing. The re-launch and post-launch support will be expensive. Spend the money. Otherwise kill the product and cut your losses.

☞ **RM2**: **Re-price high-res animator at $275.** As the reborn Animator takes off, we roll out the high-resolution animator, code named PJ, at $275. Call it "Animator Professional" and harp on the way it effectively uses 386 mode and VCR control, etc. Continue to target paint programs as well as animation, and show how POCO and user-defined inks make this the first open-architecture paint program.

☞ **RM3**: **Move animator, PJ to Windows.** As quickly as possible, we should make both the original Animator and PJ work under Windows. In the case of PJ we may have to wait for the 32 bit mode Windows, but in neither case should we wait any longer than necessary.

☞ **RM4**: **Re-price 3D Studio at $499.95.** Don't touch the product. Just move the price down to where the early adopters can afford it, including all those people who'll buy it on impulse and find out only later that this is the product they've waited for all their life. Don't discount the impact of "corporate early adopters"—at $500, lots of companies will buy one and have somebody evaluate it. At thousands of dollars, it's a major purchase decision. Doubt me? Try ordering a $3000 software package to "look it over" around here. After repricing, re-launch much more broadly. Offer demo tapes at cost through 800 number. Maybe try a little cheap-o TV. Move the product to Windows as soon as technically feasible.

☞ **RM5**: **Launch an image-processing product.** Acquire, license, or develop a product that embodies the image processing functions not provided by PJ. (Quite possibly, these facilities could be integrated into PJ or made some kind of add-on to it.) This would complete Autodesk's product line in the raster graphics industry. The product should be targeted at a $499.95 price point, and could use comprehensive manipulation of scanned images as a means of establishing its initial customer base.

New CAD system. In parallel with the work underway on AutoCAD, we should immediately begin a project to produce its successor; a thoroughly modern product that does everything AutoCAD does and provides a reasonable level of compatibility, but which is freed from the truly stupid baggage of the past. This project would target its initial product release for Windows, with subsequent platform versions thereafter based on market demand. This development project would be staffed comparable to the AutoCAD project and assigned similar priority.

☞ **NC1**: **Develop AutoCAD successor.** Immediately begin development of the next-generation CAD system—the replacement for AutoCAD. The outline, specifications, scope, pricing, and timetable for such a project are given in the "Nightmare Scenario" on page 625. The project would be chartered to take as much or as little code from existing Autodesk products as its developers wished, and to use whatever development environment and methodology they believed best suited to the task.

The initial release of this product would require a 386 or better and the 32 bit version of Windows. Other platforms would follow. The product would be able to read and write AutoCAD **.DWG** and **.DXF** files, but would not be required to maintain absolute compatibility in other aspects. For example, the command line

could be entirely absent, menus might be totally different, and AutoLisp may not be provided. This would differentiate the product from AutoCAD, so when it was introduced at a lower price with broad distribution it would not be perceived, at first, as a replacement for AutoCAD. The product would, however, serve as the base upon which such a replacement would be assembled in the future.

All existing Autodesk resellers would be authorised to sell the new product at the date of its introduction and would be encouraged to sell it to prospects they are currently losing due to AutoCAD's price. The product would also be placed into general distribution, using the channels which already carry Generic CADD, plus others we may develop in the meanwhile. If the fear generated by the existence of this product (which will be done by somebody in the next 48 months anyway, I am sure) cannot be overcome, I suggest it may be easier to take if the product is named "Generic CADD 7.0."

☞ **NC2**: **Continue product line integration.** Building on the initial release of the successor system, which will already integrate illustration, 2D and 3D wireframe, surfaces, and solid modeling in a modern application framework (providing multiple document architecture, clipboard, a full function macro recorder and editor, background generation, and inter-application hot links just as Excel 3.0 does), continue to roll in other components of the product line as pluggable pieces. Two or three releases down the road, it should be possible to paint, do image processing, create animations, and perhaps build worlds for cyberspace or spatial models for Xanadu all from the user interface (and, just as important, application linkages) of the product.

A Bodyguard Of Fears

Whenever change is suggested, particularly changes to the assumptions that have guided an organisation and are embedded in the very way it goes about its business and in the style of the products it builds, it's only human to resist, to dispute the need for change, or to characterise the changes as too risky or impossible to achieve. These objections often come neatly tagged with phrases embodying supposed disadvantages of the proposed course. I'd like to address some of these reason-killing labels head on, before they're dropped into the debate over the issues I've raised herein.

"Betting the company"

Any major change can be seen as gambling with the billion-dollar market capitalisation of the company. Yet most companies that attain great value then lose it do so by *failing* to adapt when technological progress or the market demand they change. Part of the art of running a company is making a number of small bets, each with a limited downside and unbounded potential gain. This is how you bet on the future without betting the company; rarely need a current revenue stream or key product be put squarely at risk—there's almost always a way to structure a transition so the new is allowed to compete with the old, with the market rendering the ultimate verdict as to which is better.

Since markets embody the wisdom of thousands or millions of people, and management contains but a handful, providing a variety of alternatives and seeing which succeeds allows a company to lead an industry while following the market. To do otherwise is to "bet the company" either on the judgement of a few individuals or, even worse, on the market's suddenly ceasing to evolve. You can bet the company by doing nothing.

And lose just as much.

"Fiduciary duty"

Over-cautious managements often rationalise their fear of change by appealing to their fiduciary duty as custodians of the company's assets. Deploying the company's resources, redirecting priorities, and taking actions which might have adverse short term consequences are claimed to violate the trust they exercise in the name of the shareholders.

Yet is not the duty of a fiduciary to manage the assets entrusted to him "as a prudent man would dispose of his own assets?" What is "prudent" for a bank trust officer or pension fund manager may constitute a freeway to oblivion for the management of a company that leads an industry in the midst of a technological revolution—especially if that management has signed up to deliver compounded exponential growth in sales and profits for the foreseeable future.

It's easy to become obsessed with fiduciary duty when guarding $140 million. But it's essential to never lose sight of the fact that the most fundamental and profound fiduciary duty—the ultimate act of the prudent man—is to use that money, when necessary, to protect the future of the company that earned it. Fiduciary duty is often cited as a excuse not to act.

Yet when change is required, the prudent man adapts.

"The human wave"

Suggestions that the manpower devoted to development and maintenance of our products is inadequate are often greeted with the claim that one is advocating a "human wave" approach to software development. An image of ranks of mediocre programmers stretching off the horizon is summoned up, and contrasted against our hardy band of Stakhanovite geniuses able to achieve a greatness to which no large team can aspire.

I believe this is one of the central myths of the software business, destructive of products which must, as demanded by the market, grow to such a scale that a handful of individuals can no longer do all the work by themselves. Further, it avoids facing the issues of structuring a product and project in ways that can accommodate additional human resources, including those not immersed in every detail of the entire program.

There is no indication that the user demand for additional capabilities in products is going to abate in the near future. Forget something as complicated as CAD, where it's obvious to anybody that we've hardly scratched the surface of modeling the real world—just look at spreadsheets and word processors over the last five years, and see how they've grown to meet escalating user requirements.

Although software development methodology, improved tools, and faster machines have increased the productivity of individual programmers, it just isn't realistic to expect these advances to allow a tiny group to maintain an ever-growing product. Indeed, overwork and the necessity to stay immersed in the details of the product just to keep up may cause developers in that environment to lose sight of developments which could benefit their work, or simply lack the time to step back and implement them.

Nobody is arguing for mediocrity. It's a cheap shot to assume that the work of a larger group need be inferior in quality. By subdividing a product along functional lines and parceling out work in a manner that takes advantage of individuals' expertise rather than forcing each person on the project to learn the intimate details of every corner of the product, it may be possible to increase the net productivity of many individuals.

But regardless, in a market which expects products which are far more comprehensive than those of a few years ago, surrounded by much more support material, and closely integrated with other applications and the underlying software system, the old drag racing maxim applies: "there's no substitute for cubic inches." A development team with 3500 cubic inches of neurons will, unless things are utterly screwed up, leave the lean, mean team with only 1500 in the dust and smoke, regardless of how bright the smaller team is, how many hours they work a day, and whatever the marginal efficiency and better communication they benefit from by being so few.

An ultimate determinant of how much software goes in the box as well as the quality of that software is how many man-hours were expended to create it. I believe this is a little-appreciated but important factor in the success of Japanese products. A new product launched by a Japanese company typically embodies several times the engineer-years of a comparable product from a U.S. or European company. The refinement and freedom from problems that customers have come to expect is simply a reflection of all the work that went into initial design, finding problems, and fixing them before a customer ever saw the product. Attempts to surpass the quantity and quality of a much larger team through pure cleverness and long hours seldom work outside the pages of fiction.

Learning to do things on a grander scale is part of growing up in an industry. Certainly nobody would suggest that Autodesk should try to run its business today with the staff that sufficed in 1986.

What makes us think we can develop modern software that way?

"Throwing money at problems"

The programmers' fear of the "human wave" has its parallel in the accountants' spectre of "throwing money at problems." Like most evocative phrases, this expresses both a legitimate concern and an irrational penumbra of fear. Certainly we're all aware of examples where spending more on a project only makes things worse. But that's all the more reason to treasure those rare problems which *can* be solved with money, particularly if you're a cash-rich company with the money it takes. Treasure them, but don't endure them.

Increasing the budget of a software development project in an attempt to expedite its completion rarely works. A larger group can do *more* work in a given amount of time, but additional manpower does little to compress the length of the development cycle. But if the problems one faces are insufficient staff, inadequate tools, poor production values, and a lack of promotional presence in the market, these *can* be solved by judicious spending. Enduring problems which risk the company in an attempt to conserve cash doesn't make sense when your company is valued between 20 and 30 times present earnings. Management's job is to identify the places where spending can benefit the company and apply the resources at hand to the company's needs, not to consider the bank balance more valuable than the company's future.

Conclusion

Writing this letter has not been either easy or pleasant. Researching these issues has forced me to question many of the comfortable assumptions I've lived with for many years, some dating before the formation of Autodesk. I'm aware that many may read these words as an attack on the company, or on the managers to whom we have entrusted it. That is not my intent. When I was personally involved in running the company, I was just as blind to trends outside the company, just as loath to act, and just as happy to retreat into the details of daily crises or

the minutiæ of product design to escape the larger issues facing the company. This is only human, and only natural since we are human first and managers second.

Only rarely does one both have the time and the opportunity to step back far enough from the details to see the larger picture. When that picture appears hostile to the continued existence of the way you've become accustomed to doing business, it's a shock. It takes a while to sink in, and even after it does it's hard to resist the temptation to pick at little pieces of the arguments I've presented here and justify remaining on the current course, believing that it remains the prudent, low-risk path. Being prone to such argumentation myself, I expect that many people will greet what I say here in much the same way. I urge you to weigh their words against the fundamental questions I've put on the agenda.

They will dispute this or that assertion I make in this paper. I am sure I could question just as many facts and premises in a document presenting Autodesk's current strategy, were there a strategy to document, a document expressing it, or a willingness by management to share their vision of the company's future with the individuals who ultimately must turn it into reality.

They will claim their actions are moving the company in the same directions I urge, but in a "prudent, low-risk, evolutionary manner." They will paint the course I recommend as fraught with immediate peril.

After you have heard their response, ask yourself the following questions.

Is Autodesk on the right course?

Is AutoCAD as excellent a product as it could be, given the human and financial resources of Autodesk?

Is AutoCAD receiving the marketing and promotional support that befits the principal product of a top-five software company?

Is Autodesk investing adequate resources, human and financial, to maintain AutoCAD's lead in value for the dollar?

Is Autodesk acting like a leader of an industry, seeking to create new markets and broaden the use of its products?

Are Autodesk's pricing, channels of distribution, and focus on market segments consistent with the emerging trends in the software marketplace?

Is Autodesk today what I had hoped to create in 1982—a company that can turn a great piece of software into a great success in the marketplace?

Are Autodesk's plans for new business development likely to succeed?

Is Autodesk acting in keeping with the opportunities before it, as a highly profitable multinational company with a strong position in a revolutionary industry, two hundred million dollars a year in turnover, and more than one hundred forty million dollars in the bank?

Were Autodesk to be faced with a well-funded, accurately targeted assault on the heart of its market, could the company react quickly enough to fend it off?

Epilogue: The Final Days

> We are all at a wonderful ball where the champagne sparkles in every glass and soft laughter falls upon the summer air. We know, by the rules, that at some moment the Black Horsemen will come shattering through the great terrace doors, wreaking vengeance and scattering the survivors. Those who leave early are saved, but the ball is so splendid no one wants to leave while there is still time, so that everyone keeps asking, "What time is it? What time is it?" but none of the clocks have hands.[345]
>
> —Adam Smith, "Supermoney"

One thing is beyond doubt: we are living through the final days of the original Autodesk. Whether the problems we're currently experiencing herald the waning days of Autodesk's leadership of the industry, or are merely the birthing pains of a new Autodesk, ready to accomplish as much in the 1990s as the old Autodesk did in the prior decade, will be determined in the next weeks and months.

To management I say this. The resources are at hand. The Autodesk team can and will accomplish whatever tasks you ask of them. The financial strength amassed by years of success allows overcoming any short-term barrier we may encounter. All that is needed is that you act. Act now. Act while the opportunity remains. Act while we still lead the market and the industry. Act while the customers and the market wait, in silence, for your response. Act while you still can.

If this message is understood and accepted, then the next few weeks should bring the first signs of change. In a few months, the first evidence of the new Autodesk will begin to appear in the marketplace. Things will start to get better. It will be obvious to everybody that Autodesk is again on the move, that the spirit, the confidence, and the energy that vaulted Autodesk into the front ranks of software companies in 1983 and 1984 and 1985 is now, in the early years of this new decade, carrying Autodesk further, toward leadership among the next generation of software companies.

If management does not act, this too will become obvious. It will be reflected in falling sales, declining profits, eroding market share, and eclipse of the company as an industry leader—loss of the ineffable sense that here is where the future is being built. If this happens, we will know what opportunities were squandered at the very moment Autodesk held the future in its hands, and we shall never forget.

[345] I deliberately reused this quote, which I had previously used in Information Letter 10 (see page 222) in 1983 to express the sense of urgency I felt then about the coming confrontation between Autodesk and the entrenched, well-financed, mainstream CAD companies. Seven years later, Autodesk had prevailed over the then-giants of CAD and, in doing so, had transformed CAD from an exotic vertical market application to a rather conventional software product addressing a broad variety of users and industries. The coming confrontation, now, would be between specialised CAD companies such as Autodesk and broadly-based software vendors such as Microsoft and Borland who might view CAD as simply another product within their broadly diversified portfolio of interoperable applications. As in 1983, the inevitability of this coming conflict was generally acknowledged, but proved an inadequate stimulus to make the changes required to prepare for it.

View from the Ridge

Marc Stiegler joined Xanadu in 1988 as general manager. He quickly impressed almost everybody who encountered him with his calm, competent, well-disciplined management skills—especially given that he was applying them to the Xanadu development project and team, both about as chaotic as anything I have experienced in my career.

When Ron McElhaney left on September 1, 1990, and dreading the thought of another search for a Vice President of R&D as painful as the last one had been, our thoughts turned to in-house talent. Al Green suggested Marc Stiegler to me shortly before I was going to suggest Marc to Al. I knew of no other in-house candidate willing to do the job who I considered so qualified, and since Marc was willing to do the job, he accepted, coming to work in Sausalito in early 1991.

After spending 90 days talking to people, studying the projects and operation of the department, and evaluating the situation, and right after Information Letter 14 (see page 600) was released, he wrote a series of "Views From the Ridge," distributed via electronic mail, which set forth his observations, philosophy, and priorities for the technical staff.

A View From the Ridge

by Marc Stiegler — April 3, 1991

Goodness, I now understand why VPs get reputations for being inaccessible. It is very easy to spend your life responding to the most-recent problem you've heard about. I had expected this dilemma, but I must confess I hadn't expected it to be so shockingly severe.

Anyway, I'm going to be trying to refocus, on the goals rather than the obstacles; we'll see how successful I am.

I have entitled this email message "View from the Ridge", which is where I often feel like I'm standing. From the ridge I can see 2 valleys. In the one valley there are all the people of Autodesk. In that valley, everyone is working like crazy, fighting fires, draining swamps, thrashing alligators, building a better place to live. Sometimes because of my ridge position I can see what is happening more clearly than the people in the midst of the fire; but sometimes the smoke and fog get in my way, and I cannot tell what is happening at all.

On the other side of the ridge is the second valley, a raw wilderness periodically invaded by barbarians and by tremendous forces of nature. When I look into that valley, I can sometimes see tidal waves and conflagrations working their way toward the Autodesk valley. Hopefully I will see them in time for us to prepare to meet them. Eventually, I believe it is our destiny to tame that valley as well.

I intend to give you all periodic updates on what I see, from this precarious and imperfect location, up on the ridge.

My first View from the Ridge is very brief. It is about the latest missive from John Walker.[346]

By now, I suspect everyone has heard, in one fashion or another, about John Walker's new screed, 44 close-typed pages of insights and suggestions for the future of Autodesk.

I am sure this document has received uneven circulation. Due to our grapevine's capacity for distortion, I would guess that people who have not seen it have heard amazing wild rumors about what it might contain, rumors beyond the imagining even for a science fiction author.[347]

So I have given a copy of John Walker's Information Letter #14 to Carolyn Hedrick. Anybody who wants to see what it really says is welcome to come grab a copy. Please, please treat this document as Company Proprietary.[348]

I myself am still chewing on John's comments, even though I had some foreknowledge of their nature. I am looking forward to discussing this, and many other matters, with everyone soon—this is another part of escaping from the obstacles, and focusing on the goals.

Thank you.

A View From the Ridge: Windows

by Marc Stiegler — April 7, 1991

Some of the Views from the Ridge that I wish to discuss will be controversial; fortunately, some of them will be summaries of topics upon which there is general agreement. Such generally-agreed-on points are worth noting simply to make people aware of how much agreement there is (yes, there are things people agree on, even in Autodesk `:-)`[349]

I feel a strong urge to leap into discussion some of the controversial views; but because so many of you still have not had a chance to meet me, people who are justifiably nervous about what kind of person I am, I think it makes sense to warm up with some of the simpler ones.

Perhaps simplest of all is a discussion of Microsoft Windows. For those of you who have not heard my view of Microsoft Windows, let me describe it as I have been describing it since before coming to Autodesk:

---

[346] Meaning, of course, Information Letter 14, which was officially delivered to management two days beforehand, but had been madly self-replicating throughout Autodesk in an earlier draft version the previous week, after an inadvertent disclosure of the draft. See page 600.

[347] Marc Stiegler is, of course, best known as a science fiction author. His books include *David's Sling* and *The Gentle Seduction*.

[348] Within days, Information Letter 14 was circulating freely among Wall Street analysts, at Microsoft, and elsewhere in the high tech community. Only at Autodesk was it considered proprietary.

[349] This is a "smiley," a tiny piece of ASCII art used (overused by many) in electronic mail to indicate the writer is grinning. Crank your head to the left 90°. Get it? Other smileys include `:-(` writer is frowning, `:-(O)` writer is yelling, `:-{)` smiling writer has moustache, `.-)` writer is one-eyed, `:-[` writer is a vampire, `]:-)` writer is the Devil, and `O:-)` writer is God.

When I read the trade press and its reaction to Windows 3.0, I cannot help being struck by one, and only one, analogous event in the history of computing. That event occurred at the beginning of the 80s. Everyone knew that CP/M was dead because the future belonged to 16-bit machines... but no one knew what would replace it. Then MS-DOS came out.

In 2 years there just wasn't anything else.

I expect the same thing to happen with Windows. This fills me with both horror and joy at the same time. On the one hand, it astounds me that Bill Gates could spend so much time and so much money to build an environment that is so inferior in so many ways to what Apple had 7 years earlier. On the other hand, having the windowing system standardized will allow us, and everyone in the computer industry, to focus their resources on the one environment rather than spending their effort building their own poor imitations of a windowing system for themselves.

Anyway, I have made a personal commitment to Windows: though I love my Mac, and though I find it more productive in a number of ways, I have put it in a corner at home; now I use a Windows computer for work, including writing this memo. I would rather be a master of Windows than to be one who in the end is mastered.

View From the Ridge: In the Name Of JW

by Marc Stiegler — April 8, 1991

I do not intend to make the View from the Ridge a daily occurrence. However, there's a lot of excitement/wondering/worrying going on right now. And, much as this may surprise those of you who have not even seen me yet, I have spent 90 days doing very little except listening. In the process, I have accumulated many things I'd like to say. So I'll be doing several Views here in relatively quick succession. In this View, I wish to speak of a matter that I have had disturbing indications constitutes a problem—a problem that could become very serious with the release of Information Letter 14.

I have heard many people say they agree with many of the points John's Information Letter 14 makes. I have heard a few people say they agree 100% with all the points John makes. Yet on careful listening, I am unsure whether the people who agree 100% with John actually agree with each other.

This curious twist of fate compels me to discuss the morality of citation, and the respectful use of another person's name. Citing another person, presenting another person's statements, imposes upon the speaker a grave responsibility. In the short run, uncareful citation harms the listener, who believes wrong statements for wrong reasons; in the long run, it destroys the speaker, who ceases to be believed.

Long before Information Letter 14 came out, long before I started driving to Autodesk every day, John's name was a very popular component of discussion. I guess my most favorite form of this usage is "I agreed with John when he said...." Hmmm. Let us look at the ways a person can be cited, for both moral and immoral purposes.

First, one can cite a person to give credit. This makes perfect sense if the idea you are presenting was thought of by someone else. This is a highly moral form of citation; indeed, one must give credit in this case, anything less is immoral.

Second, one might choose to cite a respected person as an eye-opener. If one is presenting a radical idea, an idea you fear might be rejected out of hand in the absence of such respect, you might think of this as a mechanism to "awaken" the person with whom you are speaking. The purpose, then, would be to assure careful attention to the reasoning behind the radical idea. On the one hand, the morality of citing a person for this reason is a bit gray. On the other hand, I must confess, I have done it myself, as often as anybody. My current view is that simply trying to arrange a careful assessment of an idea is moral, and therefore this a moral form of citation—though it places an extra burden of honor on the speaker to present the citation fully, with no truncation or spin.

Third, one might want to cite a respected person as a weapon. In this case, the purpose is not to get a fair hearing, nor to reach for a careful assessment. Rather, in this case the purpose is to create fear, to shut off assessment, to compel others to agree with no further consideration. This is, obviously, a debased immoral act, a heinous crime against both the listener and the person who has been cited. It is an attack on thinking mind, an attack that no human being deserves to have rendered against them.

The strength that makes John Walker the person he is, is not the correctness of his recommendations. It is the clarity, correctness, and completeness of the reasoning that goes behind those recommendations that is important.[350] If our company is to succeed, we must choose the clear, correct, complete reasoning, regardless of who was the initial presenter. We must judge the reasons for their own merit—not for the merit of the person who speaks, nor for the merit of the person who is cited.

As a listener, it may be very difficult to tell when a speaker wants to use a respected person's name for simple credit, or for other purposes. Fortunately, it doesn't make a difference which intention the speaker had: in the end, the listener has the choice of which message he chooses to hear. You, the listener, know that the legitimate citation is as accreditation. Treat it as such. If you are uncertain how the speaker intended it, point out to

[350] Upon reading this Brad Zehring was reminded of Tom Lehrer's "New Math" song,

> . . . And so you've got thirteen tens,
> And you take away seven,
> And that leaves five. . .
>
> *Well, six actually,*
> *But the idea is the important thing!*

I responded to Brad,

> But I don't like being compared to New Math. I feel more like the Oracle of Delphi these days. Surely you remember the Oracle of Delphi—back before Eisenhower? Anyway, in those days gods and goddesses, warriors and philosophers, poets and sportscasters would make the pilgrimage to Delphi to pose their questions to the Oracle who would, after a few tokes on some killer weed, render an answer:
>
> > "Characterising all the relevant criteria into a cost/benefit matrix and performing a cross-tabulation of correlations between free variables and results yields convincing evidence, with three sigma significance, of a causal relationship between the proposed courses of action and the spectrum of consequent outcomes within the operative time frame."
>
> Then the questioner would go off and do what they intended to do in the first place and get screwed anyway.
>
> > "Hey John, whaddya think of scrapping AutoCAD and going into the COBOL compiler business?"
> > "I think it sucks."
>
> *(Exeunt.)*
>
> (Later, in a meeting.)
> "I asked John about it, and he called it a worthy aspiration. . . ."

the speaker that you fully understand that they meant it only to give credit. Then listen to the clear, complete reasoning, and judge that reasoning on its own merit.

Visions of Visions

by Marc Stiegler — April 14, 1991

The complaint I have heard most often most recently is that there is a lack of vision, i.e., that this is the core problem in our company.

This is a complaint which I must take most personally to heart. I feel a particularly great responsibility to assist in creating the Vision of Autodesk; after all, why else would you put a science fiction writer in management? If a science fiction writer can't help to articulate a Vision, who can?

Here's the terrible news, folks: after 90 days, I still do not have a great Vision to suggest. Or rather, it might be better said that the Vision I have to share is the same one people have heard now for a long time: the vision of a Golden Age of Engineering. This vision is large enough to encompass, among other things, tools for molecular manufacture,[351] and tools to support a participative world library.[352] Perhaps this is not a satisfactory Vision any longer. Or, perhaps, the Vision, at this level, is not the missing piece so many people seek.

As an aside, it is perhaps amusing to note that you can run a very successful software company much larger than Autodesk without any Vision at all; Microsoft is the poignant proof of this irony.[353]

Still, Microsoft's failure is no excuse for us. As I said, after 90 days I have no answer; but it is fair to share with everyone the possibilities that have crossed my path in these 90 days. I also wish to ask some questions, to diagnose what people really are reacting to when they speak of a lack of vision.

Let me start, however, by talking a bit about my personal vision, the reasons why I accepted the job I am currently trying to fulfill. I think everyone can guess that this isn't a fun-filled job; only the hardest issues reach me, the issues so hard that they couldn't be solved efficiently. My life as VP at Xanadu was more pleasant—but candidly, that wasn't a joyful post either, for much the same reasons. No, I was much happier before that, when I was on indefinite leave from my former employer, close to the mountains in the Northwest, just writing.

Why did I come here?

I came here because, today there are still problems you cannot solved with either love or money; I came here to help build a future filled with products that today cannot be purchased at any price.

Neither love nor money can buy good government. Neither love nor money can buy a repair for my hip. Neither love nor money can buy the ability to live in the Sierras yet work in Sausalito. You may be surprised by this, but Autodesk is already working on products that could make the key difference in each of these areas.

[351] HyperChem.

[352] Xanadu.

[353] I can't imagine how anybody could believe this. I've met Bill Gates and many of the senior managers at Microsoft, and I think they have more vision than all the other managements of major software companies put together.

As I write this, I am looking at the January 1991 issue of Popular Mechanics. The cover story is about a vehicle, the Merlin 400, nearing completion in Davis, California. You can tuck the Merlin in your garage. Or you can motor it out on to the highway...then lift off vertically to 20,000 feet, and cruise at 322 mph to your destination. This is not a wild flight of fancy; there is a picture of the live vehicle with the article; a couple of years ago I was involved in evaluating it, and the underpinning technology for the Merlin is all straightforward, even mundane.

The story of the development of the Merlin is even longer and stranger than the story of Xanadu. As of now, Paul Moller has spent 25 million dollars and 25 years to build the Merlin. No technological hurdles remain: there are engines with the thrust, there are computers with the control precision, every aspect of the Merlin is provably within reach—and you still can't buy one. Why not?

The answer is that it is still just too hard, too expensive, and too time consuming to put it all together. Mohler needs better tools. He needs tools worthy of a Golden Age.

In 1986 the last technologies Mohler needed for building his dream were complete. With the right tools, at that point he should have been able to design the Merlin down to the last screw with AutoCAD, confirm the correctness of the design with AMEWindTunnel, confirm the safety of the vehicle in the CyberSimulator, submit the design and receive certification from the FAA through a Xanadu Information Pool, offer the blueprints and purchase the parts through the AMIX Custom Components Market, and build the first Merlin literally in his garage. It should have taken 18 months, and cost $500,000, not 25 years and 25 million dollars. If the Tools of the Golden Age were available, today we would be able to purchase a Merlin for less than the cost of some Yuppie automobiles.

Once again, this is not fantasy. There is already an existence proof: I understand the B-2 bomber, after being designed, all fit together perfectly and flew the first time. All we have to do is reduce the cost by 6 orders of magnitude. Someday, there will be a successor to AutoCAD that makes this possible. We ought to be the ones who build it.

Anyway, this is my personal vision. I wish to hurry the day when a person like Mohler can build his dream and make it available for us to enjoy.

Credits: I wish to thank Bob Schumaker at AMIX for pointing out the article in Popular Mechanics. I wish to thank John Walker for the B-2 bomber story.

View Clarification

by Marc Stiegler — April 14, 1991

The View "In the Name of JW" produced a number of surprising reactions. I was not surprised to receive surprising reactions: when sending a message to 157+ people, the chances of misunderstanding are just too great to be avoided.

One particular reaction has prompted me to expand a bit. I have been told that you could read "In the Name of JW" and become afraid to mention a corporate founder in my presence, for fear that I might misinterpret it.

Need you fear that this is now a minefield, planted by marc stiegler, that you must tread carefully around?

Management by the laying of mine fields is an interesting topic upon which I might one day write a view if it turns out to be important at Autodesk. But for the moment I think it's enough to say that I have no wish to lay mines, ever. This too would harm the open discussion needed to achieve clear, complete reasoning.

You are welcome to cite the founders with me, any time. I will always recognize the accreditation, and treat it as such. I cite founders from time to time myself; I would appreciate the same courtesy, the recognition that I too mean it only as accreditation.

Credits: I wish to thank Rob Rayles for highlighting the need for this clarification.

Valuing Corporate Image

Some folks conclude from disparaging remarks I've made about poorly done advertising and PR efforts (see pages 219 and 254) that I don't believe in spending money to build corporate image. Not at all; I believe corporate image is ultimately a reflection of the behaviour of an enterprise over time, and can benefit from a well-executed communications campaign. One often hears arguments against general corporate identity programs which claim, "It's only wasting money to stroke the CEO's ego" or "Unlike product promotions, you can't quantify the value received." In the midst of a debate about how much time and money Autodesk should spend to position itself as a broadly-based software company as opposed to "the AutoCAD company," I tried to provide a metric for the value of corporate image.

Valuing Corporate Image

by John Walker — May 13, 1991

> A cynic is a man who knows the price of everything, and the value of nothing.
>
> —Oscar Wilde

Several people have discussed ways by which Autodesk might raise its profile and alluded to the cost of advertising directed at overall perception of the company's standing in the industry. Well, it's easy to be cynical about something called "corporate image," especially when you look at how much money is wasted with no perceptible benefit in pursuit of that elusive goal. But one thing that's nice about business, especially if you're a numbers freak, is that it's often possible to calculate not just the cost, but the value of a company's image in the minds of investors and, presumably, the larger market for its products, old and new.

Suppose somebody suggests allocating some money towards reorienting our marketing communications efforts more around the company than towards promoting individual projects. What's involved would be, I suspect, more a change in focus than a major new initiative—much like the way Microsoft always refer to their products as "Microsoft Excel" and "Microsoft PowerPoint," *never* just the product name alone (even though that's what the rest of the world calls them). Like Microsoft's recent "you have an assignment" campaign, we would focus on solutions embodying several of our products, not just a single program (and as one who's spent much of the last 4 years building links between our products and prototyping multi-product solutions, I can assure you there are plenty of such stories to tell, both internally generated and crafted by customers).

We can see what it will *cost*, but what's the value? It's easy to take pot-shots at such a campaign by saying, "with such a limited promotion budget, we have to spend every cent on rifle-shot targeted product promotions, not a broader message." But resources are always limited, and still one must choose the wisest course.

As I discussed briefly in Information Letter 14 (see page 627), the price/earnings ratio of a company's stock is a direct measure of the market's perception of a company's strength or weakness.[354] Instead of an amorphous measure of sentiment, it's based on cold hard cash; it tells how many dollars investors are willing to fork over to buy a dollar's worth of the current earnings of a company. If they pay more, it's because they believe that the company is in a strong position to continue to grow in the future, thus rewarding investors with increased earnings and consequently a higher share price.

When two companies in the same business, with identical margins, bear substantially different price/earnings ratios, those numbers are the market's verdict on the strength of those companies. By promoting the company and changing its perception, one is rewarded if successful with a rising P/E. What can this mean?

Let's take two software companies with identical margins, Autodesk and Microsoft. One is perceived as a one-product applications company in a specialised industry (notwithstanding being number 4 in the PC software business overall), while the other is seen as a broadly-based vendor with fingers in system software and many applications, diversified across different platforms. As of the market close last Thursday, Autodesk was trading at a P/E of 25, while Microsoft commanded a P/E of 33 (these have tightened up since I wrote IL14; at that time the numbers were 22 and 34).

Assume for a moment that half this P/E gap is genuine, reflecting Microsoft's larger size, broader product base, and demonstrated success in opening new markets, and the other half stems from Autodesk's low profile as a software company in the general market. If, by increasing Autodesk's visibility, we were able to close half the gap, Autodesk's P/E would then rise to 29 from its current 25.

The P/E is, of course, computed directly from the share price, so that rise would imply a rise in the stock. How much? Why,

$$58.5 \times \frac{29}{25} = 67.86$$

So, the *value* of increasing Autodesk's "corporate image" to this extent would be a substantial rise in the share price. But share price doesn't give the true value either. To get that, we need to multiply by the number of shares outstanding, about 24,330,000, to get the actual increase in the value of the company. That gives us:

$$(67.86 - 58.5) \times 24.33 \times 10^6 = 227,730,000$$

or, in other words, a tad less than a quarter of a billion dollars; a number comparable to a year's gross sales receipts from AutoCAD.

Kind of puts "image" in a different light, doesn't it?

The essential part of the story that we're not communicating is that Autodesk's new product initiatives: multimedia, Xanadu, molecular modeling, the science series, and others coming down the road, are not only connected in an applications sense with AutoCAD, AutoShade/RenderMan, Generic CADD, AutoSketch, and other existing Autodesk products, but are consistent in a strategic sense as well. Every one of our current products was, at the time we launched it, on the very ragged edge of what could be done on the hardware of the time and was considered a marginal application, far from the mainstream of the PC industry. Those that have succeeded

[354] A much more detailed exposition of this may be found in Information Letter 12 on page 283.

have expanded the envelope of the applicability of the desktop computer and have created new markets which, having opened, we now dominate. This is why Autodesk products rarely get off to a quick start; they have to co-evolve with the hardware and build the applications market which they will eventually be considered synonymous with. This process can be slow and frustrating but, as AutoCAD has demonstrated, it can be very rewarding.

Were that story fully appreciated by the market and embodied in Autodesk's P/E, the value would be very great indeed.

Simulation Isn't Reality

Prompted by an article in Aviation Week*, and several comments over the preceding months about how, eventually, computer aided design would allow us to perfect products before they were ever manufactured, I wrote the following to try to inject some humility about just how much computing could help the process of engineering.*

Engineering, simulation, computation, reality, and humility

by John Walker — June 6th, 1991

Just about the time the "agenda" group[355] was created, Marc Stiegler posted a characteristically eloquent message[356] that put the gap between what we know we can do (a.k.a. "the golden age of engineering") and what we choose to put up with (a.k.a. "the golden age of management") in stark perspective.

I'm not sure whether the original message went to agenda or not. In any case, what follows stands on its own. This is in no way a rebuttal of his message, with which I concur 100%; it is an amplification intended to limn the possibilities and desirability of simulation.

There's a tendency among those of us who spend our lives building models of the real world, whether of the geometry of a mechanical part, the flow of heat around a hypersonic vehicle, or the relationship between investment in certain aspects of a business and return over a five year period, to confuse the model with reality.

The model, of necessity, must be oversimplified and stripped of much of the richness that makes the real world a better place to live than **`megalon:~uucp`**.[357] It's oversimplified not just because the computational requirements of a faithful simulation would humble even the NSA computers you're not supposed to know about, but even more because which parameters are relevant and which can be ignored simply aren't known. Consequently, when we build a model, it embodies all of our present-day prejudices about what matters and what doesn't. And therefore, we shouldn't be surprised when it fails to prepare us for the surprises that nature has in store for us.

[355] Shortly after Information Letter 14 (see page 600) appeared, Autodesk management created an electronic mail alias to which any employee could subscribe, intended for discussion of issues related to Autodesk strategy. It started out as a lively, enlightening, and useful forum for discussing a wide variety of issues. Over the months it degenerated into a meeting place for prominent airheads who posted messages of the genre, "We need day-care, better health coverage, a company cafeteria, matching contributions to the retirement plan, three month company-paid maternity *and* paternity leave, larger quarterly profit sharing, *and* higher regular salary!" I stopped reading "agenda" in August 1991 and it was abolished some time afterward.

[356] "Visions of Visions" in his "View From the Ridge" series. See page 650.

[357] The public file directory on Autodesk's main Unix server machine in Sausalito.

April 1 last, the first full-scale test firing of the Titan 4 Solid Rocket Motor Upgrade was conducted at Edwards Air Force Base in California. This is the second most powerful solid rocket ever built; only the Space Shuttle SRBs are larger. The motor ignited on schedule and performed as expected throughout the ignition transient period of 400 milliseconds.

Then, without apparent cause, pressure started to rise in the uppermost segment of the motor. In less than a second, the composite motor case burst at about 1800 psi, resulting in an explosion that destroyed the entire reinforced concrete test stand, which will cost between $20 and $50 million and take 8 to 10 months to repair.[358]

I quote from Aviation Week (27.05.91, in the library):

> Investigators determined that extensive three-dimensional computer simulations of the SRMU's firing dynamics did not reveal subtle factors that they now believe contributed to motor failure. Stirling (Col. USAF, Titan 4 program director) said the full-scale test was essential precisely because computer analyses cannot accurately predict all nuances of solid rocket motor dynamics. "That's why we test," he said.

Indeed....

As we build our models, it's wise, every now and then, to remember that they're only models—representations of the real world that are no more faithful, in all, than your Lionel train set of Christmas past was of the Burlington and Northern. Computation can, and has, given us important insights on reality which were inaccessible in the age of pencil and paper (the implications of chaos in the dynamics of nonlinear systems, for example). Yet the unquestionable triumphs of computer aided engineering which, today, are becoming manifest everywhere in products which could never have been created without first having been simulated (a few years ago, in the first flight of an airplane, the test pilot didn't even raise the landing gear; last week, on the first flight of the Dassault [makers of CATIA[359]] Rafale, it went to Mach 1.2) risk engendering a kind of computational hubris. Last October, at the Foresight Nanotechnology conference, Bill Joy suggested that instead of building the SSC,[360] we should spend the US$\$8 \times 10^9$ to build a computer which could simulate all the results the SSC could create. (This proposal evoked, at least at my table—deep among the jackal bins—derisory hoots.) But consider... if everything in Heaven and Earth is programmed in **`HORATIO.SYS`**, *pourquoi pas*?

Because reality is richer than simulation, and experience beggars computation. Richard Feynman said:

> For a successful technology, reality must take precedence over public relations, for nature cannot be fooled.

Albert Einstein said:

> Subtle is the Lord, but not malicious.

[358] Following repairs to the stand and modifications to the motor to eliminate the most likely causes of the explosion, the second test firing went perfectly.

[359] The mainframe CAD system, marketed by IBM in the U.S. which I constantly used to show just how little of the functionality of a genuine CAD/CAM system AutoCAD provided.

[360] The Superconducting Supercollider particle accelerator. After US$2 billion had already been spent on construction, mostly for digging tunnels at the site in Texas, the U.S. Congress terminated the SSC in September 1993.

Kelvin R. Throop sez:

> Nature is a tricky Mother. Reality must take precedence over *everything*, or you'll wind up in deep doo-doo.

We can happily spend the next 20 years developing faithful simulations of the physics discovered in the last century and creating thereby tools which will empower people to create products we can't begin to imagine today. But let's remember we're merely bottling a vision of the real world, not the real thing. Reality always asserts itself through surprises, and that's part of what makes life worth living.

The New Autodesk

The first four recommendations I made for turning around Autodesk in Information Letter 14 implied reorganising the company around product line profit centres rather than overhead functions such as Development, Marketing, and Sales (see page 635). The management response to Information Letter 14 began with a top-to-bottom reorganisation of the company into "business units" focused around principal product lines. Unfortunately, management's response to Information Letter 14 basically ended with the reorganisation. On June 16, 1991, Marc Stiegler, who had served as the chief architect of the reorganisation, laid it all out in a paper titled "The New Autodesk," accompanied by a 39 page booklet of organisation charts. Reading between the lines with the benefit of hindsight, one can see how this plan contained the seeds of its undoing. The U.S. domestic sales department and all the overseas territories remained independent of the product organisations (due to intense lobbying by the managers affected). The product organisations still could not, therefore, shift the attention of the sales organisation away from easy AutoCAD revenue onto emerging products and markets. The structure also reinforced the passivity of senior management, removing it one level further from a position of overall leadership and instituting an "Executive Review Board," whose major function was to adjudicate squabbles between subordinate organisations, not to plot the grand strategy of the company and see that it was implemented, as senior executives are supposed to do.

Overall, the organisation Marc proposed was not inherently flawed and was certainly an improvement over the "Old Autodesk." The reality was, though few people had yet realised it—certainly not me—that no *structural change in the company could turn Autodesk around without a change in the way senior management led the company, or a change in personnel at the top. With the right people, it could have worked. With the wrong people, it was foredoomed.*

The creation of the Americas territory in December 1991 returned many of the marketing functions to a central overhead organisation, effectively undoing the essence of this reorganisation.

The New Autodesk
by Marc Stiegler
June 16, 1991

"Now it's time to consider whether Autodesk has become stuck in the past; whether it's time for Autodesk to change and how. The changes may be unpleasant to contemplate and difficult to carry out, but they may be just as necessary and, if successfully made, as rewarding."

— John Walker, *Information Letter #14*

There never has been, and probably never will be, a "best" organization for a company with hundreds of employees. There are only organizations which are better for a particular company for a particular period of time. The restructuring along product lines proposed here is intended to serve Autodesk better during the coming decade would simple growth-by-accretion on the current structure.

This document describes a framework for product line oriented structures, as well as a proposal for how the framework could accommodate the new Autodesk. It starts by discussing four problems that the new structure might solve; continues with the goals of the restructuring; and then goes on to discuss the organizational components of the new structure and the functions of the people in it. The document ends with a set of questions and answers; some of the answers duplicate material in the document, but they also help explain why the proposed restructuring might successfully solve additional problems.

Some people may agree with the idea of moving to a product-line structure simply because John Walker proposed it in *Information Letter #14*. Some people may agree simply because this organizational structure has served other companies, such as Microsoft, so well. Some people may never agree with it until it proves that it works. For most people, however, the key question may well be, "Why does a product-line structure work so well for companies like Microsoft, and why would it be better for Autodesk?"

To partially answer this complex question, I list here several current and near-term corporate problems that a product-line structure could solve.

Problem 1: Overhead

Currently, virtually every department in the company is an overhead center, from programming to technical writing to marketing to manufacturing. Eventually, all such arrangements breed inefficiency. When swift shipment and low cost are of secondary importance, as they are in overhead centers, the entire department's focus slowly shifts from doing the job quickly and effectively to doing it *perfectly*, with decreasing regard for how long it takes or how much it costs. Worse yet, the meaning of "perfectly" also shifts, away from a customer's definition to one that reflects the priorities of the overhead center.

This is not an indictment of the people who make sincere efforts to make the system work—it is an indictment of the system itself. The Federal Government is the shining example of this: there are many excellent people who work grueling hours in the government yet cannot point to any worthwhile national achievement. Over the long run, such an overhead operation is the breeding ground for slowly deteriorating effectiveness.

At Autodesk, the Tech budget for FY 92 is 43% larger than Tech costs were in FY 91. Yet it is difficult to determine where the money went. No one can look at Tech and identify a 43% growth in productivity or effectiveness. On the contrary, almost every product seems to take longer to deliver than ever. One can look at each project and explain why it's having problems; one can also look at each expenditure and say why it hasn't paid off with more products yet. But if one does this for every project and every expenditure, one winds up suspecting that there may be a forest behind all those trees.

By moving more activities from overhead centers into product lines with direct profit-and-loss responsibility, we redefine success and failure in customers' terms—the most effective way of

consistently leading and winning in any marketplace. Even if we believe that such a redefinition is not needed today, it seems likely that it will be critical to our success in coming years.

Problem 2: Responsibility

With the current structure, neither the success nor failure of any product can be credited to anyone. This has peculiar ramifications in the case of failure: suppose the manager of a software development project hires such a large staff for a project that, when the product ships, it does not generate enough revenue to pay for the staff. The marketing people are not to blame: they sold the product well and brought in lots of money. From their point of view, they succeeded. Meanwhile, from the software manager's point of view, he/she also succeeded: the product shipped on time, with all the features customers could want.

Certainly, the software manager and the marketing people wanted the product to succeed overall; they are sincerely interested in the company's success. But when push comes to shove, they also must consider first that part of the puzzle for which they are responsible. Here, every individual was a success, yet the company failed.

By moving responsibility for profit and loss into the hands of a product manager whose total focus is the total success of the product, we ensure that someone takes credit for its success... or its failure.

Problem 3: Responsiveness

Arranging to do an AutoCAD port to a new platform (even to "just another UNIX Box") requires a tremendous amount of consensus-building. The VP of Marketing cannot initiate a port on his own—he needs the support of the VP of Tech. Similarly, the VP of Tech cannot initiate this action on his own either—if the VP of Marketing won't support selling it, it hardly makes sense. The lowest level person in the company with all the authority needed to do a simple port is the President.

By giving the AutoCAD family a general manager with authority to initiate marketing and development, we eliminate one or two levels of management overhead. Achieving a consensus of many participants will remain important, but giving GMs authority eliminates many potential decision-making bottlenecks.

This is a very important improvement, since decisions need to be responsive to rapidly changing market conditions. Breaking out the AutoCAD family into multiple products, each with individual profit-and-loss responsibility, could reduce even more management overhead for many decisions. By making product managers responsible for profit and loss, we give them a better yardstick for success than any higher manager could supply. (And, at the same time, we get higher management out of the role of trying to act like market forces themselves).

Problem 4: Planning

Microsoft's multi-year product plans have been lauded for their excellence. What does it take to do this kind of planning?

First, it takes both marketing and technical people, standing toe to toe, figuring out what the product means. It takes a leader who is totally focused on the long-term, overall success of the product. And it takes incentives for the leader and the marketing and technical people to put it all together into a plan. The product manager is the leader: in the product-line structure, he or she has the resources needed to develop a plan; the negotiation with the Executive Review Board over development and profitability targets (described below) is one of the incentives to develop it.

Goals of the Restructuring

At the highest level, the goals of this restructuring are:

Making the company more responsive to opportunities in the marketplace.

Making the company more effective in strategic planning.

Giving diversification efforts the level of attention they deserve.

These goals are achieved through the following supporting goals of the restructuring:

Moving authority and accountability closer to the actual work. This makes the company more responsive to the needs of customers.

Shifting the company's focus from technical or marketing success to the life-cycle success of a product. This facilitates strategic planning for the product, and improves responsiveness by boosting the chances that the response will be effective. By focusing on the life-cycle success for new products from their inception, we also improve the chances for the success of new products, which is critical to diversification.

Guaranteeing that there is at least one person who is dedicated to the comprehensive success of each individual product. Again, this improves planning, responsiveness and the chances for the success of new products.

Removing upper management from the details of product planning, freeing them to focus on corporate strategic planning and selecting opportunities for diversification.

Improving communication between technical and marketing people on individual products; this improves responsiveness.

Giving people the authority to succeed, which again improves responsiveness.

Types of organizations

This proposal identifies the following three types of organizations which can comprise most parts of a product-line oriented company:

Business Units

Service Centers

Corporate Functions

Business Units

Business units generate revenue; the individuals responsible for the business unit are accountable for its long-term profitability. In a homogeneous environment like the U.S., the product lines are the most obvious examples of business units, although there are other organizational units, such as foreign subsidiaries, which are business units as well.

Service centers

Service centers supply services to the business units and bill back the costs. There are at least three reasons for having a function provided by a service center rather than by the business unit itself.

First, the level of service a business unit needs may vary significantly over the course of its work cycle. It would be inefficient to have the business unit hire a full-time staff capable of handling a peak load; it makes more sense to staff the service center with people who can move from project to project as needs change.

Second, providing the service may entail special equipment. If the equipment is expensive and is not used to full capacity by any one business unit, it doesn't make sense to have other business units duplicate its purchase.

Third, the service may maintain a corporate standard. The reliability with which the standard is maintained could be increased by having all the business units share the same service center, which would specialize in completing the work in a way that maintains the standard. If a service center exists to maintain a standard, it should publish a guideline document which describes how business units that cannot use the service center (for example, physically remote business units) can maintain the standard.

In each case, the advantages of having a business unit use a service center must be traded off against the advantages of having dedicated members of the business unit do the work.

Service centers and business units would usually interact as follows. The business unit needs a service. The service center estimates how much it will cost. If appropriate, the two organizations negotiate exactly what service is actually provided and what it actually costs. The business unit gets billed for actual costs accrued by the service center.

One attractive aspect of a service center is that the agreed-on costs negotiated with each business unit justify the service center's budget to acquire additional resources. As a result, questions attempting to justify percentages of budget on a macro scale (e.g., "What percentage of the R&D budget should go to QA?") are replaced by smaller, simpler questions. In addition, the direct billing gives the business unit the incentive to use the service center wisely.

The person in charge of a service enter may, in practice, hold a title such as manager, director, or VP; in this document the generic term functional manager, or FM, is used regardless.

Corporate functions

The corporation as a whole picks up those functions which cannot be provided by business units or service centers. For these functions, it's either difficult to make billing a meaningful incentive to use the service (as with finance and accounting—it's not clear how to meet legal requirements for accounting if every business unit

contracts an outside accounting firm), or so dangerous to the company if business units *don't* use the service that there's a need to eliminate all disincentives for using it (as with legal, as discussed later).

The number of corporate functions should be minimized. This is where bureaucracy grows, since the incentives for keeping lean and mean are so weak. If someone comes up with an interesting idea for how to turn a corporate function meaningfully into a service center, we should go for it.

Product Lines

A product line includes most people who work full-time on a product, including that product's programmers and marketing people. Since there are quite a few projects at Autodesk that could conceivably become separate product lines, it makes sense to group them into product families. Each product family would have a General Manager (GM) and a General Management team. Each product within the product family would have a Product Management team, with the Product Manager (PM) reporting to the General Manager. The General Manager would report directly to the President of Autodesk.

In the simplest case, the General Manager would have the following responsibilities and authorities vis-à-vis product managers:

> The General Manager negotiates with the Product Manager on the product's target profitability, taking into account the development needed to position the product and the family for long term success.
>
> The General Manager may break a single product into multiple products, or merge multiple products into a single product, as he/she deems appropriate for strategic reasons.

In this simple case, the GM and the PM do not need daily contact, as most of the authority resides with the PM. In practice, this relationship will often be built on a product-by-product basis. For example, if a product is allowed to run at a loss, because the GM sees a strategic advantage to the product family, the GM and PM probably need to work out a relationship which very clearly guarantees that the GM's strategic purpose is being fulfilled.

Also, because in the simple case the GM does not have much work to do, GMs would frequently double as PM for one of the products in the family. As a rule, no one should act as PM for more than one product—the purpose of the product-line structure is to guarantee that the total success of each product gets the total dedication of at least one person, and having one person act as PM for multiple products defeats that purpose.

Tentative Product Lines and Business Units

The following are proposed assignments of products to families:

AutoCAD Family

The AutoCAD family could include the following products:

AutoCAD itself
AME
Autoshade

This is only one possible breakout: the General Management team for the AutoCAD family has the authority to make the final breakout decision. At this time, the breakout of AutoCAD into products is so undecided that the breakout listed above does not match in any recognizable way the functional chart which accompanies this document.

In creating product lines, one usually wants to break a family into as many independently profitable products, with individual accountability, as possible. However, just because a product can have a separate price tag doesn't make it an independent product. A good example would be an ADS application that generates IGES files. Such an application could be priced separately—but its profitability is irrelevant to its success because IGES's main purpose is strategic, to help AutoCAD penetrate markets where IGES is required.

One could also make a case that Autodesk sells AME and/or Autoshade for strategic reasons. If these products get bundled with AutoCAD, their profitability no longer matters, and they should cease to be independent product lines.

In the end, the General Management team will make these decisions.

Multimedia Family

Multimedia is already broken out, at least at the GM level, like a product family. It would include Animator, Animator Pro, and 3D Studio. One interesting question is which product family the Science Series product line should belong to; this is one candidate.

Retail Software Family

This would include Generic CADD, GenCADD, and AutoSketch. This family is also a candidate as the home for the Science Series.

Molecular Modeling Family

This would include all the Hypercube products, including Hyperchem, HyperNewton and HyperNDO.

Information Family

This would include Xanadu and AMIX.

Other Business units

The only other organizations that clearly should be business units are Autodesk's foreign subsidiaries. Other organizations that could become business units are noted in later sections.

Service Centers

There are several new types of service centers needed with a product-line structure, and there are interesting questions as to which current Autodesk departments become service centers, which remain as corporate functions, and which get incorporated into product families. Before answering these questions however, there are additional features of service centers that deserve discussion.

General Concepts with Service Centers

Billable Versus Non-Billable Activities

Exactly what does the business unit get billed for when it uses a service center? There are parts of a service center that might reasonably be left as overhead, i.e., not billed to the product lines.

Before running through the possible distinctions between billable and nonbillable items, we should review the purposes of making a service billable. One purpose is to encourage the service center to be more efficient in its handling of business unit requirements; the other purpose is to encourage the business units to use the services efficiently, for example by preparing materials in a way that the service (whether it is an in-house service or an outside service) can cost-effectively digest.

Given these purposes, an exact accounting suitable for the SEC is clearly overkill.

Should the cost of the computers and other capital equipment used by the people in the service center be billed back to the product lines? In this proposal, the answer is *yes*: this becomes the basis on which the functional manager justifies his/her capital equipment budget.

Should the basic facilities costs be distributed across the employees of the service center and then billed to the product line? The proposed answer is *no*: it is not clear that it adds to the effectiveness of the incentives, and is easier for accounting to track if left out.

Many of the service centers have a certification aspect to their activities, including as a minimum the development of guidelines for product lines that are unable to use the service center. Should the cost to develop such guidelines and certification activities be billed? The proposed answer is *no*: this is geared toward a corporate goal. The organizations that most use the service center are the ones that least need the guideline document; it would be very distorting to the incentives to bill the users of the service to support the nonusers of the service.

If we create a team inside the service center to improve the operation of the service center, should this cost be distributed and billed? The proposed answer is *no*: once again, such work contributes to corporate long-term goals, not immediate product line goals. The success of such efforts should be measured based on milestones and end results, as with other development projects.

Should the cost of the manager of the service center, be distributed across the employees for billing? The proposed answer is that this be decided *on a case-by-case basis*, depending on the extent to which the manager of the service center works on corporate goals versus individual product goals (i.e., what percentage of the time does the manager spend worrying about guideline documents and facilities as opposed to nurturing specific relationships with product lines?).

Every service center has a number of odd cases which should be reviewed for operation as overhead or as a billable service. The functional managers should develop proposals for the Executive Review Board (described later) for deciding on the exact breakout.

Review of Services

How do we determine if the service centers are doing a good job? The answer may cause a moment's anxiety for members of service centers who have come to believe that bitter adversarial roles are a necessary part of their lives. The good news is that there are examples in the corporate world of places where such relationships are rare exceptions: it is possible to run companies larger than Autodesk with less friction and fury.

Product lines and business units will be measured according to how well they serve their customers, by measuring profitability. It is appropriate for service centers to be measured according to how well they serve their customers as well. Consequently, this proposal recommends a yearly review of the service centers by the GMs and PMs.

This does not necessarily portend disaster for the service center. Just because one GM or PM is disgruntled with a service center, that is not the end of life for the center, because there are five other business units involved in the review. If all the PMs and GMs do not have a common perspective on a service center, then what is indicated is not the censure of the center, but rather a more detailed analysis. Furthermore, if the vast majority of PMs and GMs are unable to tell whether they're getting good service or not, the company has a much more fundamental problem than the breakdown of communication between product lines and service centers.

There are additional aspects to the review of a service center, including how well the service center maintains corporate standards and how well the service center maintains guideline documents.

Meanwhile, this is not the only forum for communication between a service center and a business unit on how work is progressing. In general, if the GM or PM has a problem with a service center's service, the GM or PM should seek out its director/manager to discuss it. This works the other way, too: the functional manager can seek out a GM or PM if he/she perceives a strained relationship.

NonUse of Service Centers

If, after the GM has talked with the FM, there is a business reason why the service center cannot effectively meet the needs of the GM, the GM should present his/her alternative plan to the Executive Review Board for acceptance. The plan should include the GM's strategy for maintaining any relevant corporate standards.

Guideline Documents and Authority

As noted earlier, service center should create guideline documents for business units that cannot use their services directly. With the coming of guidelines, an obvious question arises: who has to get permission from whom to disregard a guideline?

The FM of a service center can alter or break the guideline without permission: he/she is, after all, the expert.

The more interesting question is, does a GM or PM need to get permission from someone to disregard a guideline? Here, we should probably define a new term to distinguish between two concepts: a "guideline" is a recommendation which a GM can disregard, accepting the consequences

of the action, without consultation; while a "policy" is a rule which a GM must get authorization from the Executive Review Board to disregard.

The purpose of the restructuring is to grant the "authority to succeed" to PMs and GMs; therefore, guidelines are, by and large, the more appropriate mechanism for the company.

Does Autodesk have issues which require policies to resolve? Just in case there are, it makes sense to define a mechanism for creating policies.

Guidelines are created by the service centers. Policies, however, should be created by the Executive Review Board. People in service centers can propose policies to the Executive Review Board—but the Board must make the decision. Before anyone creates a policy, the Board should scrutinize it to make sure that it states the underpinning requirement in the broadest possible terms, rather than as a particular implementation strategy, to maximize the policy's flexibility.

Since a GM can disregard a guideline, it behooves the maintainers of a guideline to list the reasons why the guideline is a good idea. The strength of these explanations will often determine whether the guideline is followed or disregarded.

When a GM disregards a guideline, he/she should be aware that the service center may no longer be able to assist the GM with matters related to the disregarded guideline, i.e., once the process expressed by the set of guidelines is broken, the director/manager of the service center may not be able to support other pieces of the now-broken process.

New service centers:

Corporate Marketing:

This organization will be responsible for coordinating marketing between product groups and with our international subsidiaries. They will have responsibility for market intelligence and analysis. Most important, they will have responsibility for correctly positioning Autodesk as a broad-based software company through advertising, public relations, development of collateral, trade shows and other promotional events. They will be responsible for coordinating consistency and worldwide standardization of corporate style and packaging.

Developer Services:

This would be a service center of programmers who work on different projects at different times for different product lines. In our current organization, there are times when Dev Test and Build groups act in this capacity, building components for products whose programming staffs are overcommitted and have some tasks which can be done with relatively little spin-up by an extra person.

This service center would also serve a load-balancing function during a product's long-term life-cycle. Because individual products are managed by their own profitability, and funded by their own revenue streams, it is possible to imagine "quiet times" in the development of a product. For example, when a product is released, it may make sense to shift expenditures from development to marketing and shrink the development team during this period. Rather than lay off people who have just achieved a great success, or attempt to hastily find new homes for them, these people can join the developer services center and explore projects with a variety of product lines.

This service center would not be expected to be 100% committed to product lines all the time. As a result, Developer Services could, in its spare time, undertake short projects that might become new products. There seems to be a boundless supply of ideas at Autodesk for objects that could go on bonus disks, or into the Science Series, or into new market areas. Developer Services could develop into a place where these ideas get explored.

Developer Services would be expected to have most of its people employed with the product lines most of the time.

Where should Developer Services report? There are at least three interesting possibilities: 1) the VP of Technology; 2) the same organization as training and support, since short projects are also undertaken by training and support when time permits; or 3) Advanced Technology, which should, in principle, be where the largest number of radically new ideas crops up. This proposal puts Developer Services into the Advanced Technology organization, for the potential synergy with radical ideas worthy of exploration.

There are some challenging questions regarding the operation of Developer Services. For example, once an exploratory project has gotten under way, if a product line needs the services of a person working on the exploration, who gets priority? Clearly, at some point in its evolution, an exploratory project becomes important enough to be worth completing, at least in comparison with some lower priority projects in the product lines. One of the first tasks of the FM for Developer Services will be to come up with a set of guidelines for making this tradeoff

Existing Organizations Proposed As Service Centers

In this proposal, technical publications would become a service center. Though some of the product lines may employ a number of writers full time, the requirement for writers and for publication production varies over the course of the product life cycle. The publications group also supplies a level of format standardization that gets regular praise from our foreign subsidiaries.

The Build group could conceivably become a service center. If the GMs were to decide to continue to use the Build group as it stands, it would continue in its current form, with the addition of a bill-back process so the GMs and PMs account for how much money they've spent on builds. If most of the build work were to move inside the product lines, it would make more sense for the people remaining in the build group to move to Developer Services, from which they could sell build services along with other software-related services. Even if the Build group were to be completely dismantled, it would be necessary to develop a guideline document for the build process, and to maintain a "certifying agency" that checks to make sure that source and object files track one another, for QA purposes. This certification agency could be independent, run from Developer Services, or what could be more appropriate, the certifying action could be performed by the QA organization, since QA is the group for which the certification is being performed. This proposal suggests putting the build certification process with QA.

Dev Test has a similar relationship to becoming a service center: if the PMs wish to use Dev Test services through a service center, it would continue in its current form. Otherwise, some Dev Test people will go with the products that they work on full time, and the others will join Developer Services.

This proposal assumes that many of the people of Dev Test and Build will go into product lines, in which case the rest would be merged together into the Developer Services organization.

QA is a most difficult case. One can argue that QA should be an overhead function, not a service center: if a product goes out the door with serious flaws, it reflects on the whole company, not just on an individual product. Therefore, it is in the best interest of the company to eliminate any disincentive for using this service, such as direct costing to the product line. On the other hand, such a quality black mark is not as damaging to the company as some kinds of legal problems: as the Microsoft Word 3.0 fiasco demonstrated, you can ship an utterly vile product and still retain your reputation for being the superstar of the industry, indeed without even noticeably impairing your premier status in the market.

It is also possible to argue cogently that QA should be a part of the individual product line. The General Manager's incentives with respect to individual product are essentially identical to those of the President of the company: they are focused on the long-term welfare of the product, and suffer from neither a technical team's natural inclination to ship sooner rather than later, nor a QA team's natural inclination to ship later rather than sooner.

On the other hand, breaking up the QA process would reduce the application of QA standards across the company; and on the third hand, breaking up the QA process might result in some interesting innovations in QA methodology.

The middle course is to keep QA as a service center while we see how the rest of the restructuring plays out. This proposal recommends keeping QA as a service center. Our current QA department supports a number of processes other than software testing, which is the service center part of QA; these other activities may be more appropriately operated as overhead operations. The FM of QA should come up with a proposal on this matter, as discussed as a general need for FMs earlier in this paper.

Technical support would become a service center as it is already divided into product lines.

Training could conceivably become a business unit. This, however, would involve making a change in the corporate strategy, since a profit-motivated training center would want to compete, in some sense, with our distributors. For the moment, this proposal recommends making training a service center as well.

Sales could fit any of the three categories of organizations.

> Whether thought of as a corporate function, or viewed as a business unit. the underlying dynamic for the sales group is the commission structure. Commissions take sales outside the breakout into business units, service centers, and corporate functions.
>
> Sales could also be made part of individual product lines. Not having sales as part of the product lines could disenfranchise the PM and GM from a very important part of the process of getting the product into the customer's hands. On the other hand, sales is also the critical point at which the company and all its products are presented to the distributors and customers as a unified family. Certainly, even though sales is kept together as a single organization, it would make sense to have a structure internal to sales that can parallel the product families (i.e., have people in sales devoted to selling each product family).
>
> This proposal recommends making sales a service center. This would allow the PM to trade off marketing expenditures with sales expenditures. Having a structure inside sales that parallels the product families would facilitate its setup as a service center.
>
> This proposal further recommends keeping the retail sales organization at Generic separate from the sales organization in Sausalito, although this situation warrants further review.

General procedure for moving people from service-like existing departments to the new structure For some people, there may be multiple, functionally similar locations in the new structure for them to move to. An author who works nearly full time on AutoCAD, for example, could reasonably continue in that role either as a part of the tech pubs service center, or as a member of the AutoCAD product family.

Here's a proposal for how these negotiations should be carried out:

> The product manager chooses whether to make an offer to the individual to join his/her product full time. The individual then has the choice of joining the product line, or staying with the service center. There is one point the individual should be aware of if he/she chooses to stay with the service center: the product manager has the authority to hire a full-time person to undertake the individual's existing tasks inside that product line. If the PM does this, then the individual will of course find his job altered, since the PM will have effectively hired a replacement for the existing job.

Corporate Functions

We are not going to be able to eliminate all overhead activities, at least not without new innovation, beyond all known organizational strategies. The remaining corporate organizations at Autodesk would include all activities which clearly have an important purpose, but for which there is no clear way to make the accounting either effective or useful.

Proposed corporate functions include:

> Accounting and Finance
>
> MIS
>
> New Business Development
>
> Advanced Technology
>
> UNIX Network System Administration
>
> Personnel
>
> Operations
>
> Legal
>
> It might be possible to make our legal department a service center. However, a legal error by a single product line can have such terrifying consequences for the entire company, it makes sense to eliminate any hint of incentive to avoid using these services. So, the legal department sneaks in as a corporate function.
>
> Later, it might be worthwhile for us to explore splitting legal functions into corporate requirements which serve as corporate functions, and other functions which could become billable services. However, this proposal makes all of Legal a corporate function.
>
> It would also be possible, because of its anti-piracy efforts, to treat legal as a business unit. However, this also creates a number of odd incentives on the participants, so this alternative has been rejected until someone makes a really cogent case for it.

The maintenance of the UNIX network by one group, and the 3Com network by another, will get more exciting as we move marketing and technical people together. This relationship will require review as we move into the restructuring. For the moment, however, this proposal leaves MIS and the UNIX Network System Administration groups alone.

Interactions Between Organizations

The Executive Review Board

The Executive Review Board is composed of all the executive officers of the company. This board negotiates with GMs the mix of profitability and development activities required to maximize long-term success.

When a product family misses its targets, either for profitability or for long-term development, the Executive Review Board reviews the product family to ascertain the cause of the problem. Outcomes may include changes as simple as lowering profitability expectations, and as significant as terminating the product family or replacing the general manager.

The Executive Review Board will have quarterly reviews with GMs; ideally, these meetings would be held just before the quarterly Board of Directors meetings.

Guideline for Product Line Goals and Milestones Negotiation

The normal process for setting product family goals would be:

The GM team prepares a plan that includes key development milestones, U.S. profitability targets, and worldwide unit sales targets. This plan is presented to a group which includes the Executive Review board, the other GMs, and the FMs. The other GMs have as an explicit review purpose the task of identifying opportunities for powerful integration across product families.

If a GM identifies an opportunity for powerful integration, a team is established to assess the cost/benefit of that integration (in the simple case, this team is just the GM who identifies the opportunity and the GM who is presenting the plan). If the Executive Review Board decides the benefits to the company are great enough, the profitability targets will be lowered to accommodate the additional development cost needed for the integration.

The Executive Review Board and the GMs also have an explicit purpose to identify efforts which are being duplicated across product lines. When such duplication is identified, the GMs and the Review Board must resolve the duplication. In complex cases, the resolution will require the development a corporate strategy; the team to create the strategy would include participants from each interested product family.

Often, the seeming duplication of effort across product lines may be appropriate, due to real differences in needs and goals. At other times, it could be appropriate to cancel one of the efforts, levying upon the surviving project the requirement to meet the needs of the other product families: this solution is really viable only when the requirements for the product families are very similar. Finally, a combined team may be brought together to build a solution viable for all the product families. When R&D is required, such a combined team could as a default be placed in the Advanced Technology organization for the duration of the development effort, though other reporting structures are possible. There would be a good incentive on the GMs to use such a shared solution, since the development costs would not come out of their budgets.

Weekly Meeting of Al's Staff

There will be a weekly meeting of Al's staff, which includes the GMs among others, to work on conflicts, discuss processes, and identify strategic issues. Each week one GM will give a more complete review of activities and changes in his/her organization.

Tact as a Component of Management

Surprising as this may be to some people, concepts of tact were invented to increase the clarity of communication: by using tact, one can avoid arousing defensive emotions which inhibit successful understanding and agreement.

I would like to present two examples of communication between a product line and a service organization, since this is where communication is most at risk. And, I would like to look at both tactful speaking, and tactful listening—there is art to tact, at both ends of the communication.

Example 1:

The GM or the PM runs into the service organization and says, "I must do X," where X is a significant change from the plan as of the day before.

Often the truth is a bit more subtle. A statement that is both more truthful and more tactful at the same time is, "We have a set of requirements which, if we do X, will be fulfilled." It is probably worth a thousand misunderstandings for the GM to take two seconds to consider whether the second phrasing is more effective.

Communication is a two-way street: if the GM uses the first phrasing (perhaps because he/she has been working too hard and is not quite up to par), it will often be to the advantage of the team for the person in the service center to ask about, or perhaps even assume, the second phrasing.

Example 2:

The FM of the service center runs into the product line offices and says, "You can't do Y," where Y is the thing the GM or PM most desperately wants to do. Often a more correct and more tactful statement would be along the lines of, "Here are some issues with doing Y. I am interested in working out the issues with you." It is worth a few seconds for the FM to choose a clear and correct, but perhaps ever so slightly tactful, phrasing.

Once again, the listening GM has the latitude to understand a poorly phrased comment by a harassed FM in a way that does not need a strong emotive reply. The GMs really will have the authority to do what they really need to do; assertions that they can't do Y will not succeed as vetos; they need not fear someone will prevent them from succeeding. They can afford to respond with politeness and tact, since they won't need baseball bats to be effective.

General Management Team

A general management team must have expertise in each of the following areas: technical development, marketing, organizational behavior, personnel and financial responsibility. It will be rare to find a single

individual with all these forms of expertise; as a consequence, the General Management team may include individuals in addition to the General Manager him/herself.

A reasonable analogy here may be made to the Release Management teams used at Autodesk and other software companies with great success. An AutoCAD release has both a manager and a chief engineer, working largely as coequals. The manager and the chief engineer are able to work effectively as a team not only because they get along well with each other, but also because there are reasonably natural lines of demarcation for deciding which partner should have final authority over each issue. Also, each manager/chief engineer pair works out the lines of demarcation in a customized way, depending on the exact set of strengths and weaknesses of the individuals.

In a similar way, General Management teams would work out lines of demarcation of the total authority that fit well with the complementary forms of expertise supplied by the team members.

Two styles of team seem natural: A General Manager accompanied by a Chief Marketing Officer, and a General Manager accompanied by a Chief Technical Officer. We may wind up with other combinations too, focusing on combinations that achieve all the objectives with a minimal number of team members.

Criteria for assessing a GM

Did the GM achieve the development goals set at the beginning of the year, whose purpose was to position the product family for successful growth in later years?

Did the GM's products work effectively as components in meeting corporate goals, i.e., did the products work well with other products, did the marketing and documentation and quality augment the public's awareness of Autodesk as an excellent software company?

Did the GM meet the profitability targets in the US?

Did the GM meet the unit sales targets worldwide?

Contractual Authority

For signing checks, the GM should have the same authority as a Vice President. The GM should be able to sign contracts for services whose cost is as great as the GM's check signing authority can pay for.

Product Manager, Obligations and Authority

The product manager controls the development and marketing budget for the product. Except in special cases as noted with respect to the GM, this individual has the authority to hire, fire, promote, and generally do anything necessary to get a product into the hands of the customer.

A particularly challenging question is: To what extent does the PM have the authority to disregard the service centers of the company, to perform those services either inside his/her product line, or to go outside the company for that service? This question has a multiple part answer. First, the corporate service centers are effectively subsidized, simply because their space, heat, and electricity are all paid for out of overhead, so the PM has a

real profitability incentive to use the internal service. Second, the PM does have an obligation to make his/her product complement other Autodesk products in terms of marketing, documentation, quality and integration. Hence, using a service center which is set up to maintain corporate standards will usually make life easier in achieving that goal. On the other hand, the PM really does need the authority to succeed, not the excuse to fail. Consequently, the PM can get released from the need to use corporate services with an articulate explanation of the reasons for the GM and the Executive Review Board.

Criteria for assessing a PM

The criteria for assessing a PM are essentially identical to those of a GM, with one addition: because a PM has many people working directly for him/her, there is an additional question with respect to the management effectiveness: How positive and enthusiastic is the morale of the employees? Since most GMs act as PMs for one of the products in their product family, this will typically be a part of the review of the GM as well.

Responsibilities of VPs of Marketing and Technology

The Vice Presidents of Marketing and of Technology have the following responsibilities:

Review the plans developed by the GMs for strategic growth, to facilitate the negotiations between the Executive Review Board and the GMs on profitability targets (for example, if the VPs concur with the GM on major development efforts needed for long-term success, they would recommend to the Executive Review Board that they relax the profitability needed from the GM to offset the costs of the development).

Seek out opportunities for strengthening the corporation by integrating products across product families (i.e., they would do for the independent families what the GM does for the products within the family).

Seek out opportunities to reduce duplication of effort by business units by initiating corporate development efforts.

Seek out holes in the product family mix which should be filled with new products or product families.

Oversee projects still under development that have not yet been assigned to a GM.

Identify new strategic directions for diversification.

Seek out ways to improve the overall structure of Autodesk.

Review of Employees in a Matrix Organization

Employees of service centers spend much of their time working for other people in other organizations. To perform a review of such an employee, the line manager in the service center should talk with every manager for whom the employee has worked since the last review, and use that input in the evaluation. As with the annual review of the service center as a whole, if a single manager is disgruntled with the work of the employee, that

should not spell disaster for the employee; indeed, if there are widely conflicting opinions about the employee's performance, this is an indication that something more complex is happening that needs to be investigated.

Questions and Answers

How will we arrange for integration across product lines?

There is no simple answer to the problems of improving integration across product lines. Actually, there are reasons to believe that such integration might work better in the new structure than in the old one, in contradiction to the appearance of a stronger split between the products. Before attempting to prove that the new structure will work better, however, it is worth reflecting for a moment on how well integration works across products now, to see how difficult it will be to at least maintain this level of quality in integration.

In the current structure, Autosketch, Autoshade, and AutoCAD programmers all sit very close together, all tied organizationally through a VP of Software Development. Surely, this is an environment that would foster tight integration if any could. Yet even the most generous spectator would have to say that the integration among these products is weak. It would seem that organizational boundaries are not the problems most fundamentally responsible for poor integration; the difficulties lie elsewhere.

The new structure has a number of incentives for improved integration: simplest is the profit motive for the GMs of being able to cross-penetrate the customer bases built by the other product families. The part of the new structure that may produce the most interesting results is the creation of a body of people whose major focus is on product integration: the Executive Review Board. The members of the Review Board will focus more attention on these matters than ever before, simply because so many other responsibilities have been removed from their shoulders in the new organization.

How do we straddle the line between Research and Development?

A classical question for all high-technology companies is: What is the relationship between research and development? Here the question can be phrased: What avenues does the new structure offer for different kinds of research to support different kinds of development?

The new organization has at least three different ways research and development can cross. There is the Advanced Technology organization: in addition to developing wholly new technologies, the Advanced Technology group also plays a role in developing technologies that need to be shared among product families, as described earlier.

A GM can also start an in-house R&D effort in support of the milestones and goals agreed upon with the Executive Review Board.

Also, Developer Services can undertake short experiments which can be singled out for further development when they bear interesting fruit.

One of the problems in our current organization is that, to start something new and different, you need a massive level of consensus, including in principle the VP of Tech and the VP of Marketing. In the new structure there will be fewer people you need to get consensus from, and in some cases there may be alternative routes to achieving the same goal because there may be several different GMs you can interest in the idea. For example, you could initiate a pen-oriented development effort by convincing either the PM of AutoSketch (who could

include a pen-oriented enhancement in his/her next release), or by convincing the GM of the Retail business unit (who could start a new project, perhaps by splitting the AutoSketch line into two lines), or by convincing the GM of the Information business unit, or by convincing the sequence of people needed to start the project in Advanced Tech. All of these organizations become centers of action, rather than centers of veto power.

How do we improve communication between groups?

No matter what structure you build, there are groups between which communication is difficult: you can't put everyone in the same building, and you can't put everyone in the same room for every meeting.

The new organization will improve communication between marketing and technical people who are working on the same product. This communication has been in disrepair for such a long time, it seems like a most urgent need in the company. It may well be that, in another eight years, it will be time to reorganize again to repair the communications which are less effective in the new organization (which at that time will be "the old organization").

Doesn't this company already have too many rules and regulations inhibiting operations? Isn't the idea of having yet another bunch of guideline documents the antithesis of a successful company?

Actually, it is difficult to tell whether the company has too many rules and regulations. They aren't written down, and as a consequence, not only is it easy to make up new ones on a regular basis, it is impossible to tell how much the new ones have complicated the system—you can't even see how thick the stack of pages is to ascertain how much pruning we need.

Guidelines can give people power as well as taking it away, particularly when the guidelines explain why one approach is better than another. There are companies in the world that have policy manuals and are still dynamic. There is risk in writing guidelines; but there is risk in leaving it all unwritten, as well.

What if the number of policies (as opposed to guidelines) begins to grow and become an impediment? Doesn't this always happen eventually?

Since the policies will require Executive Review Board approval, they are not likely to grow in number very quickly. Indeed, by the time they grow to the point of being an impediment, it will probably be time to restructure the company again anyway.

If they grow faster than expected, we'll be aware of it because the policy book will get thick. At that point we can implement additional impediments to the growth of policies, things such as sunset laws, and requirements that at least two members of the Executive Review Board be able to quote all the policies from memory, and that any policies forgotten in quotation are stricken from the books.

If one product family has a down quarter, what effect does that have on other product families?

It must not have any effect on other product families. One could construct a scenario in which it would be best for a company to restrain the product families that are on plan when a key product family has trouble; but not Autodesk, not now.

Items to watch for future modification

The entire new structure should be watched carefully for opportunities for improvement; however, there are some specific areas that have already been identified which are worth watching closely to see whether change is appropriate. These areas include:

Coordination of the UNIX network system administration with the MIS 3Com network system administration. This will get more complicated during the time of the move, since these networks will need to get even more deeply intertwined. Hopefully better technology can resolve some of these problems.

Coordination of corporate marketing and technical publications. Each of these organizations create material for the product; it needs a coordinated look and feel.

Coordination of the retail sales group at Generic and the dealer channel sales group in Sausalito.

Keeping all of Legal as overhead, or identifying components of the legal activity which should be charged back to business units, service center style.

Making training a business unit.

In a completely free market, if a business unit went to an outside service and demanded extremely short-notice turnaround, the outside service might well charge a premium price for the extra effort. Since our internal service centers are nonprofit organizations, this kind of "premium charge" does not make sense. Yet the absence of such a premium charge mechanism distorts the incentive system: to the PM or GM, it looks no more expensive to demand service on an hour's notice than on a week's notice, even though it may very well be more expensive—such sudden requirements cost us in terms of scheduling conflicts and angry frustration. What would be an appropriate mechanism for repairing this distortion?

Acknowledgements

Many, many people have been involved, in various degrees at various times, in the development of this proposal. Below is a listing of all the people the author can still identify at the end of the process, who had specific unique suggestions and insights that caused the author to add or improve specific passages of the final draft. All ambiguities, omissions, and errors, are solely the responsibility of the author.

David Ang
Carolyn Aver
Moonhie Chin
Ruth Connolly
Malcolm Davies
Scott Davis
Al Green
Scott Heath
David Kalish
Volker Kleinn
John Lynch
Tom Mahood
Sandra Marin

Pete O'Dell
Kern Sibbald
John Walker

The New Autodesk: June 28, 1991 Change Pages

by Marc Stiegler

Below are listed the major changes, additions, and clarifications in the June 28,1991 version of "The New Autodesk":

Service Centers

There was a graphic which depicted some organizations as service centers, some as corporate functions, and others as a mix of service center and corporate function. In fact, under close inspection, every "service center" has corporate functions as well as service functions. As noted in the text, the Functional Manager for each service center should present a proposal to the Executive Review Board breaking out which activities should be handled as a service, and which should be handled as a corporate function.

Service Center Guidelines

There are 2 kinds of guidelines that service centers may—or may not—need to do. One kind of guideline is the guideline describing the service center's process, so that remote organizations can use the same process even if they can't use the service center. Only organizations that have a certifying function need to prepare such a guideline. Those are the centers that uphold a corporate standard of some type.

The other kind of guideline is a submissions guideline, i.e., what should materials look like that the PMs bring for service? If a PM wants to break such a guideline, the appropriate response is not to tell him he can't do it, but rather to tell him how horridly expensive it will be to him if he does it that way.

There are organizations for which it really doesn't make sense to do either of these kinds of guidelines, such as the sales group, as an example.

Service Centers Serving Other Organizations

Organizations other than business units may use service centers; once again, the costs get billed back, and the organization which plans to use a service center should allocate money in their budget for using such services. An example is Advanced Tech, which needs the service centers in much the same way the product lines need them.

Definition of Matrixed Employees

Not all employees of service centers are "matrixed" to product lines. A "matrixed" employee is one who conceptually joins the product group, usually to the extent of sitting in their office space, while working with the product team. Sales is again an example of a service organization wherein the employees are not matrixed.

Problems with service centers that do not matrix employees are conceptually problems with the service center as a whole, and should be addressed to the FM.

The Sales Service Center Budgeting Process

In the New Autodesk, the way the sales department budgets changes only slightly. To understand the change, first let's review very briefly how sales did their budgeting in the Old Autodesk:

- Upper management and sales would develop a revenue target;
- Sales would create a budget that could achieve that target;
- Upper management (in particular, the VP of Marketing) would review the budget.

Two changes occur. First, the revenue targets are broken out by product line. Second, for each product line, the Product Manager plays the role of "upper management" in the discussion of targets and in the review process.

The Library

The Autodesk Tech Library does a lot of important activities that are atypical of a library, such as database searches. The Library changes its name to Information Resources, and becomes a part of Advanced Technology. It operates as a service center (with some corporate functions).

Japan Group

The Japan group will become a part of the AutoCAD product family. Today, this is where the vast bulk of their work resides. In the long run, it may be more appropriate to have a separate service center for the Japanese ports of our software, to ensure that emphasis is not skewed by the organizational structure. This arrangement will be under continuous scrutiny to assure that it works.

The Science Series

The Science Series will remain with Advanced Tech until we have identified a Product Manager and fully reviewed the choice of business unit where it should be placed.[361]

The Corporate Marketing Group

The goals of the corporate marketing group are to:

- enhance strategic focus on customers: targeting, requirements, emerging opportunities;
- assist in establishing balance between internal product development and external market realities;
- facilitate integrating both marketing planning and programs across all product groups and regions;

[361] The Science Series was subsequently transferred to the Retail Products Group, then terminated in 1993.

- create a consistent corporate image as umbrella to integrate all internal Autodesk marketing activities.

Corporate Marketing will include both Marcom[362] implementation, and a strategic focus on planning.

Training

Where "The New Autodesk" discusses training, the particular kind of training being discussed is dealer training.

Check Signing Versus Expenditure Authority

GMs do not need check-signing authority, which is the mechanical part of the purchasing process. What they need is expenditure authority. The General managers will have the authority to approve expenditures for their business unit at the same level as an Autodesk Vice President.

Incorporating Customer Feedback

How do customer requirements make it into products? This was a most popular question. The document "The New Autodesk" does not address it directly.

Indirectly, the new structure creates an environment even more conducive to incorporating user feedback than the Old Autodesk. Because of the closer ties that the PMs and GMs have to profit and loss, they have a stronger incentive to care even more about the customer's desires. This in turn means stronger incentives to establish strong lines of communication with those who are close to the customer (notably, the sales people). Meanwhile, the sales organization will have a dedicated point of contact for each product line, and that person will have a strong incentive to communicate as well. Finally, the sales people themselves will retain the strong incentives to communicate that they already have (namely, to have a product they can sell more of more quickly). Consequently, any successful GM/PM will establish forms of communication well suited to the needs of that product, both with our own sales people and with other sources of good customer information.

Areas For Continuing Assessment

The 2 areas of most active concern in the discussions have been the coordination of the sales groups and the coordination of the 3COM and Unix mail systems. We will continue to assess these areas for improvement during the transition and beyond.

[362] Autodesk-speak for "marketing communications"—in essence, an in-house advertising agency.

Choose Wisely

After reviewing the reorganisation of Autodesk around product lines proposed in Marc Stiegler's "The New Autodesk" (see page 659), which seemed to me, on the whole, pretty reasonable if a tad fussy, it suddenly dawned upon me that the reorganisation was the entire *response to Information Letter 14 (see page 600), and this dismayed me profoundly. I had recommended a reorganisation (albeit much less ambitious than the one Autodesk was embarking upon), as a means to the end of drastically increasing the investment in our products—both in development and marketing and sales. It now seemed clear to me that the management was going to roll out this reorganisation, shuffle boxes on the org chart, then go on precisely as before—neglecting the products that brought in the money. It was a fact, for example, that two and a half months after Information Letter 14 was published, the number of programmers working on AutoCAD had actually dropped from the level cited in the letter. I wrote these two memos to Marc Stiegler, as the member of senior management I felt I could most effectively communicate these concerns, to urge management not to squander the opportunity they had to genuinely turn the company around. Re-reading them in 1993, I think they're a little intemperate, but that's a reflection of how distraught I was with what appeared to be a betrayal of Autodesk's future. These memos had no effect, as far as I can determine.*

From: John Walker, Tue Jun 18 22:28:43 1991
To: Marc Stiegler
Subject: New, Improved Autodesk

Thanks for the copy of "The New Autodesk" paper,[363] which came to hand today, and for the acknowledgements therein. I've just finished reading the document, and I have a mixed reaction. I want to share my impressions with you, because I think they may be useful in understanding how the plan is received within the company and, more importantly, in charting the actions which will occur in the days which will follow. Having been born during the presidency of "give 'em Hell Harry," before the advent of tact, what follows may be more "old Autodesk" than new. But, here goes.

First, the plan itself. I have no objections whatsoever to the goals, structure, or the people chosen. The plan looks fine and should have a good shot of achieving the ends for which it was created. Certainly, once the dust has settled, things should at the very least be better than before, if for no other reason than that accountability has been implemented for many functions which were buried in the overhead previously, as you so clearly pointed out.

[363] See page 659.

So why am I not ecstatically happy or recharged with enthusiasm to plunge in to this reinvigorated organisation? That's a little more complicated, and it's only very indirectly related to the details of the plan itself. The issue is one of morale and direction, and it's there that I think immediate action is required in order to avoid losing the momentum that's conferred by a period of change. I think there's a risk that the reorganisation plan will be viewed by many as exalting minute details of management structure over directly addressing the genuine problems of the company and its products. I share this worry myself. First, when the IL14[364] bomb burst, I believe and I said at the time that management missed a truly golden opportunity to turn around the morale of the company. I think that by appearing defensive and reactive rather than aggressively seizing the initiative, the perception of the very problems I outlined in the letter was reinforced.

In a real sense, it was Autodesk's latest (and I hope, final) PR disaster (and do not doubt that it was a disaster—both my personal reaction to the employee meeting I attended and the comments that people addressed to me after the week of meetings[365] fully justify that evaluation in my opinion). First of all, at the time the meetings were held, almost a dozen specific steps had been taken to turn around the company and arrest the drift—things which included the Windows project, the unified 80x86 release, the CD-ROM, the newsletter, and others which had been set in motion well before I raised them. Yet these steps, which could have been used to impart a feeling of renewed momentum were not even *mentioned* in the meeting I attended, nor if they were raised in other meetings, didn't capture the attention of those I spoke to. Instead, reorganisation appeared to have been seized upon as something management could do which would defer addressing the truly difficult issues and, in a way, delegate the ultimate responsibility for the choices that must, eventually, be made even if by making no choice at all.

I have nothing against the reorganisation, after all I put it at the top of my priority list, and I approve the plan but as I said in the E-mail I sent shortly after arriving in Switzerland by which I hoped to squelch the grognards for a while and buy time for the implementation of the plan, any reorganisation whatsoever does nothing at all to address the fundamental underlying problems that the company has gotten itself into. It only creates conditions under which, one hopes, the problems can be more effectively diagnosed and successfully corrected. Now that the new structure has been chosen and implementation is underway, attention must turn to dealing with the substantive rather than the structural—to getting Autodesk back into a position of leadership in the industry.

While the reorganisation was underway, the general reaction was to wait and see what happened. OK, we've waited and we've seen. And now the question is, "what next?." People are waiting, and they won't wait forever. I don't mean that mass resignations are in the offing, but what's at risk is losing a once in a decade opportunity to get things moving again. What, in my opinion, has to happen within the next week, and certainly within the next fortnight, is for senior management to lay out a crystal clear plan for where it is leading the company in the next year, the next five years, and in the long term. Whether this plan comes from negotiation between PMs, GMs, and ERBs, whether from the mind of an Executive Vice President, or from the UN High Commissioner on Refugees doesn't matter so much as that the plan is unambiguous, ambitious, but achievable (Autodesk desperately needs a clear-cut success at this point).

"The New Autodesk" chartered by the plan is a creation of management. What is needed to make that structure a success of leadership, and it is leadership, explicitly from the top, which is now sought by all the people who have waited to see what was being made of their company. The problem, which has been alluded to in the

[364] Information Letter 14. See page 600.

[365] In the week of April 29, 1991 meetings were held with all Autodesk California employees in which Al Green, Marc Stiegler, and Malcolm Davies presented a preliminary form of the contemplated reorganisation as their reaction to Information Letter 14. I attended one of these meetings shortly before leaving California for Switzerland. My reaction, confirmed by most of the other attendees I spoke to afterward was "Is that it? Where's the substance of the change?"

guise of "vision," "direction," and other fuzzy words is real. And it needs addressing immediately.

I founded this company, and I've worked here for about a quarter of my expected professional working lifetime. And yet I cannot answer, with any degree of certainty, any of the following questions:

- What do the senior managers of Autodesk *believe in*?
- Where are they leading the company?
- What do they see as the ultimate goal of their strategy?
- How do they define Autodesk—not as in the mission statement, but in its current and near future role in the industry?
- What will be in Release 12 of AutoCAD, and when will it ship?
- Will there be further releases of AutoCAD thereafter?
- How will Autodesk address the widely-perceived aging of its principal product?

Here's the point. If I worked for Microsoft, Apple, IBM, or NeXT, in any capacity, I wouldn't have any problem answering any of those questions—the mission of those companies is evident and is clearly articulated both inside the company and to the outside world. Even if I worked for Lotus, this would be clear—I might disagree with where they were going, but I'd know where it was. When people here talk about a lack of vision, I don't think they're worried about a lack of imagination of what the future can hold; rather it's a sense that since no near- and medium-term strategy and tactics to implement it have been enunciated, there probably aren't any. And that rightly concerns people.

I sometimes think that Autodesk suffers from an inferiority complex which may date all the way back to my decision to build a leading edge company from a core of maintenance programmers. Consider this: when Steve Jobs contacted Autodesk to present his "vision of the future of CAD," in short order Autodesk had rented Marin Vets' and turned out hundreds of people to hear him. Yet when has Autodesk's senior management presented *its own* vision of the future—overall, of CAD in general, and of AutoCAD in specific? If I'd been running the company (which, thank Almighty God I'm not), I would have called a similar all-company meeting within 48 hours of the circulation of IL14, and I would have walked onto the stage alone and explained where I was leading the company and how we were going to get there.[366] That would have, in my opinion, if pulled off successfully (and that isn't hard), bought at least a year's worth of high morale and productivity and would have erased any negative impact of IL14. Lost opportunity #1.

Now, what you need to do is seize the second opportunity. Do the same thing—invite everybody, walk out there without notes, without slides, and without cutoff time and lay out the whole thing. The 21st century... how Autodesk can be bigger than IBM in 50 years... how Xanadu, AMIX, Hypercube, and AutoCAD fit together... why what's happening in the industry requires changes in Autodesk... what's going to be in Releases 12, 13, and 14... why we're building a new CAD system... and precisely what the milestones are, how we'll achieve them, and how we'll be rewarded as we do.[367]

Then, I'd make the whole thing public. Why not, it's going to get out anyway. I'd buy a page in the WSJ and print a transcript, then run an ad every month with a checklist as the milestones were accomplished.[368]

[366] This was the genesis of the idea for the all-company meetings Volker Kleinn and I held at Marin Vets' in December 1991 and January 1992.

[367] This is a rough outline of the agenda we used in the all-company meeting of February 1, 1992. See page 718.

[368] This is the genesis of the product checklist slide I first showed at the special shareholders' meeting in New York on January 30, 1992 and, with shipped products checked off, in every public appearance I made afterward. See page 724.

Just as in the recommendations in IL14, you salt the whole thing with easily achieved goals in the near term, so there's a sense of progress and accomplishment that blows away the sense of stagnation and malaise. Then you trumpet each success along the way, and before long the higher hurdles seem easier and easier.

Or, you can delegate, wait, plan, and co-ordinate. I think it is time to lead. People have waited, and now they're ready to get to work. They want to know what to do, and why.

General Patton, who I don't necessarily admire but who I find myself continually quoting, regarded morale as a consumable item of logistics—just like petrol, food, or ammunition. He budgeted a substantial fraction of his time to visiting the front, touring hospitals, and speaking to the troops precisely because he believed that by doing so he kept the reservoir of morale filled. Morale is an intangible quantity, but it has dramatic consequences which show up before long on the balance sheet. And I think that here at Autodesk, now in June of 1991, it's time to fill 'er up.

I fear, reading this over, that I haven't made myself as clear as I'd like to. It's hard, writing, with no sense of feedback as to whether you're getting through. But I feel a sense of urgency about this that's caused me to drop everything else to write this message and to send it now without waiting to see if a rewrite would help clarify the essence. I feel there is a truly singular chance here to turn the company around, not just by infusing people at the bottom with a sense of purpose, but by causing them to see their management in a new light, and that to pass it by would be to forfeit much of the potential you've worked so hard to create in forging the structure for the New Autodesk.

You've made a singular and widely-circulated offer to listen, and that's wise and wonderful, as wisdom comes from many minds. But I think that first and foremost, leaders must lead, and to lead they must speak. Ahead I see challenges so great that I do not know how we can achieve them. In 12 to 18 months I believe that we must ship a version of AutoCAD than runs on Windows and OS/2 which is upward compatible with the current product, but which includes a user interface that blows away Ashlar Vellum and Corel Draw; it needs to provide a bottom to top revision of the basic core facilities of the products—colours, text fonts, image import and export, ellipses, etc., etc., so that our $4000 product isn't humbled by the likes of the new Mac Draw. And I don't know how to do all that in such a short time. But I think that if you ask that kind of miracle of the right people in the right way, then give them what they need and keep them going, they'll pull it off. They never let me down.

At least that's the view from scenic Switzerland as the dusk descends, here between the cement factory and the train tracks.

It is amusing to note that Autodesk stock hit its all-time high for the ten-year period 1983–1993 on the day I sent Marc Stiegler the above memo about squandering opportunities. Afterward, as the consequences of inaction became manifest, the stock declined relentlessly, finally bottoming 8 months later at $23\frac{1}{2}$ after the bad quarter was announced (see page 718).

```
June 18, 1991   1:30  CLOSE

ACAD       Yesterday's  Last      Changes/      Volume/
            Close        Bid       Asked         Date
              61        60 1/2    - 1/2        229,000

ACAD stock is down 1/2 from yesterday's close

The Dow Jones Average of 30 Industrials is down 7.15 to 2986.81
```

After re-reading my first "salvo over the pole" memo the next day, I decided a second strike was in order. The first memo clearly identified the problem, but didn't spell out the consequences of continued inaction. I spent most of the next day drafting the following MIRV memo and lobbed it Sausalito-ward as dawn was breaking in California.

From: John Walker, Wed Jun 19 16:45:10 1991
To: Marc Stiegler
Subject: New, Improved Autodesk

After reading over the message I dispatched last night about the importance of immediately laying out Autodesk's strategy in the aftermath of the reorganisation, I'd like to clarify a couple of points that I don't think I made adequately.

Further, I didn't indicate that the message, although in reply to your mail, was in essence directed to executive management as a whole. Please feel free to share any of these thoughts with other members of the group.

Whose hand on the tiller?

My message gives the impression that I'm dumping the entire responsibility for communicating Autodesk's strategy on you, personally. That isn't my intent. People look to senior management to chart the direction of the company. The person who *needs* to enunciate that strategy, whose fundamental job it is to do so, within and without, and whose doing so would most effectively put an end to the sense of stagnation is Al Green, the CEO. If he can't, or won't, then Malcolm Davies, in his role as Executive VP, and from his focus on communicating the company's message to the outside world should do it. If he can't, or won't then you should do it. It isn't enough, by a long shot, to say "be patient…Ruth and John[369] will be in touch." The issue isn't the plans they're developing, but rather what you, the senior management, are asking of *them*—and that resources you have allocated to them to achieve those goals.

[369] Ruth Connolly and John Lynch, co-general managers of the AutoCAD Business Unit.

If nobody can, or will, specify this then within days the impression will gel, probably irretrievably, that nothing has changed—that people can go on doing what they were doing before, working on the same diddly projects, slipping schedules and redefining specifications at will, and there is not to be an immediate and fundamental change in the way work gets done at Autodesk.

What is at risk is the entire program of renewal I called for in Information Letter 14, which was tacitly accepted by management in meetings with employees inside the company, although, based on my conversations with analysts and reporters over the last couple of weeks, largely poo-poohed in comments made outside the company.

How much, and when?

Do you, or do you not, intend within 12 months to commit development manpower and promotional resources to AutoCAD which are comparable to those which Microsoft, Lotus, Word Perfect, or Ashton-Tate would commit to a product with similar revenue, margins, market penetration, and importance to the company? What are your specific plans, both in terms of staffing, funding, and schedules? What existing resources will you divert in order to achieve these plans, and what new resources do you intend to add, and when?

These are the substantive issues of IL14, the resolution of which a program of reorganisation was undertaken to address. I, along with the rest of your shareholders await the statement of your plans to address these widely-perceived problems. I, along with the rest of your employees await the clear direction we need to know what we should be working on, and the schedules we must achieve to carry out the company's programs.

If, in fact, it is your intention to continue the development of Release 12 with a staff which has actually shrunk substantially since the completion of Release 11, to fail to address the obvious, glaring holes in AutoCAD compared to virtually every one of its competitors, to assemble unambitious releases and then slip them from month to month, then I must tell you that is my intention to call you (the senior management as a whole) to account before the shareholders and ask you to explain why you believe such (in)action is justified and in the best interests of our company.[370]

Possible and impossible.

The core problem, inside and outside the company, is a belief that "it's impossible," or "those guys in Sausalito will never be able to do it," or "that would take a totally different way of running the company." Well, OK, maybe it will. But don't tell me that with 165 people in software development and 140 million in the bank you can't deliver a product that has functionality and ease of use comparable to products from start-up companies with staff measured in the tens. Or that it's acceptable for the company's flagship product to acquiesce, release after release, to user interface standards which would not be even momentarily contemplated in products developed by smaller, less richly funded, and more peripheral efforts from our own company such as Tapestry[371] and AMIX.

[370] This marks the instant when I first began to doubt the capacity of the individuals in senior management of meeting the challenges involved in restoring Autodesk's competitiveness. As the months passed with no substantive changes being made, this doubt grew into a certainty that wholesale changes at the top were required. On September 26, 1991 I articulated this in my memo "Stating the Problem" (see page 691).

[371] A project underway in the Advanced Technology Department which had something to do with a user interface for Xanadu. I was never able to figure out what it actually was, despite reading a number of documents, seeing a demo, and talking to several of the developers.

Do the people working on AutoCAD today know, and believe, that their lives are about to undergo an enormous change—that the standards for performance and the ambitiousness of what is undertaken in the project, and the scale of resources committed to it are about to jump by a factor of from 5 to 10? I don't think so, at least from the contact I've had with them recently. And if not, why not?

Is it because there is no such intent? If that's the case, you are continuing down the path of neglect of the product and eventual extinction of the company's revenue stream.

Is it because they haven't been told? If that's so, then the very people you most need on your side have been excluded from the turnaround in the company's course at the very moment you must get them on board to have any hope of success—they're the people who have the knowledge of the product that's essential to transforming it so, like it or not, they're the ones you have to work through to achieve any near-term results.

I wrote Information Letter 14 in an effort to set in motion the changes I, and many other people, believe necessary in order to achieve all the wonderful things we know this company has the opportunity to accomplish. In it, I handed you a tool to remake the company in any form you felt best, and I have thrown my full support behind that effort. And now I await the results. Be assured, I will not countenance continued drift.

Whether you have in me an enthusiastic contributor or the worst nightmare a corporate management can have: an articulate, wealthy, major shareholder acting in the interests of the other shareholders and in keeping with the goals for which he founded the company, asking of management in public simple questions for which they have no answers, will be decided in the near future.

Choose wisely.

Added Value

The notion that the software business is unlike most other industries and offers unprecedented opportunities to those who grasp this fact pervades this book. In 1988's "The New Technological Corporation" (see page 459) I tried to explain this in detail, but few people seemed to really "get it."

In September 1991, London-based The Economist *published a striking confirmation of Autodesk's uniqueness, ranking it as the most successful company on Earth in terms of added value—the difference between the costs that go into a company's products and what customers pay for them. Autodesk's high added value is a result of technological leverage which adds intellectual content to its products.*

The following is Autodesk's press release, issued a month after the issue of The Economist *containing the study. Note the 1991 "fear of greatness"—Autodesk came out Number One worldwide, but the release only mentions this in passing.*

October 7, 1991
For Immediate Release

The Economist Magazine Names Autodesk One of the World's Most Successful Companies

SAUSALITO, Calif. — A recent article in *The Economist* magazine names software leader Autodesk as one of the world's most successful companies, Autodesk, Inc. announced today. See "The Best Companies," *The Economist*, September 7, 1991.

According to a study conducted by the magazine in conjunction with the London Business School (LBS), Autodesk ranked number one in the world based on a new measure of corporate quality: added value as a percentage of sales during the period 1981/82–1990. Some 2,000 of the world's top publicly traded companies were included in the study. Autodesk's average added value during the period under review was 33.9% of sales. Added value "was typically 5%" for the rest of the field, the LBS/*Economist* team reported.

The study contends that the concept of "added value" is a better measure of corporate performance than other profitability indicators inasmuch as added value "measures how much more a firm's output is worth than all its inputs of materials, labour and capital." As such, the measure provides a better guide for shareholders, states *The Economist*.

To arrive at a company's added value, the study took its operating profit, adjusted it for depreciation and subtracted its capital charge. According to the LBS/*Economist* team, added value as a percentage of sales is a measure that reconciles most of the difficulties encountered in company performance comparisons such as differences in the cost of capital and accounting practices.

Stating the Problem

As summer gave way to fall in 1991, I grew increasingly exasperated with the inability of Autodesk to come to terms with the challenges facing it. Since I moved to Neuchâtel in May of 1991, I'd had impromptu visits from John Lynch and K.C. Jones, Al Green, and Malcolm Davies, all seemingly willing to listen, but for one reason or another, unable to act upon the issues we all agreed were vital to the survival and continued success of Autodesk.

When I wrote Information Letter 14 (see page 600), the thought never entered my mind that we'd need to replace the senior management of Autodesk—all they needed was some education in the realities of the contemporary software market, and they'd be able to continue their five-year unbroken success-streak which was, after all, longer than my stint as Autodesk CEO.

As the months passed and the only evidence of change at Autodesk was a "reorganisation" that re-drew the TO but didn't seem to change the priorities of the people in the boxes, I began to think more and more about whether the only way Autodesk could escape the downward spiral I believed it to be in was to find new leadership at the top. On September 26, 1991, I wrote the following letter and circulated it to a very short list of people whose discretion and judgement I trusted entirely, to see if I'd gone off the deep end in reaching this conclusion. I was not alone.

After deciding that a change a the top was required, I started working out how it might best be accomplished. Unexpectedly, as I was pondering ways and means to bring about change, Al Green contacted me to say that he'd decided to retire and initiate a search for a new CEO. The coincidence may be difficult to accept, and conspiracy buffs will never believe it, but that's what happened—I guess Al and I reached the same conclusion at the same time.

My respect for Al Green and what he accomplished at Autodesk (see page 697) was reinforced when he, as I had in 1986, concluded that Autodesk would be best served by handing the tiller to a new skipper.

Stating the Problem

by John Walker — September 26, 1991

Back when I was teaching myself engineering out of the Junior Woodchuck Manual, one of the first things I learned was that often the hardest single step in solving a problem is simply stating it correctly.

Recently we've all been involved in discussions about the state of Autodesk, post re-org,[372] and I think it's safe to summarise them by saying that we all believe that serious problems remain unaddressed. Before we start considering what might or might not be done about them, let me see if I can state the problem as clearly as possible, and then let's see if we agree on it. If so, we'll sweep aside lots of irrelevant stuff that might otherwise occupy our attention—matters which are symptoms of the problem rather than the problem itself, and which can best be addressed by fixing the cause, not palliating the effects. I find that my reaction to the seemingly accelerating flood of "zingers" which create in me such states of anger and frustration that I am barely able to do my work has a tendency to obscure the problem which I now believe is the First Cause of the individual events. Over the past three weeks I've been trying to get a handle on the nature of the underlying problem.

So here goes.

The problem is executive management.

Autodesk doesn't have any.

The people who form what is now called the Executive Review Board are genial, well meaning, and for the most part hard working individuals. But whether from lack of capacity or failure of will, they are not behaving as executives. By failing to provide the leadership Autodesk needs, they are depriving our company of a resource which is generally recognised to be one of the principal factors which determine the future of any business.

Autodesk's executives are behaving as supervisors or managers, not executives. There's a difference, and it's crucial.

A manager carries out plans, accomplishes projects, and implements processes that are defined for him by his superiors. A good manager will show initiative in identifying opportunities in his area worthy of pursuing, but will present such proposals to the executives to whom he reports, then defer to their judgement as to the merits of each proposal and whether it is consonant with the overall strategy of the organisation. A manager who consistently identifies new ventures which eventually succeed is spoken of as "executive material."

Autodesk's executives are not executive material. Present them with a clear description of a task: "Find a campus site," "Reorganise around product lines," "Diversify into molecular modeling," and they will usually get the job done. That is management, not leadership. It is not what executives do. Executives must *identify* the tasks which the company should undertake, *select* the managers to accomplish them, and then *provide the leadership* and support to ensure that the tasks are carried out as they envisioned.

When have Autodesk's executives ever done this?

Even the name "Executive Review Board" betrays a passive, reactive view of the rôle of the executives. I think it's perceived as something more like a congressional appropriations committee than the leadership team of a company which intends to extend and expand the enviable record of growth in its first decade into the second and beyond.

Their mistaken conception of the rôle of an executive is not something Autodesk's management is inclined to hide—in fact Malcolm Davies' recent "refutation" of my Information Letter 14 in Cadence[373] vividly demonstrates it. " 'The Windows project was already on,' says Davies, one of those at Autodesk who does call the shots." Indeed, and who would doubt that it is the most important strategic software development project in the

[372] The July re-organisation of Autodesk. See page 659.

[373] "The Autodesk Reorganization—Walker Speaks From The Mountain Top" by Kathleen Maher, Cadence, September 1991, page 28.

company. Who initiated it? Which executive first identified the opportunity in Windows, then later the risk of being too late with too little in the market? Which executive decided to divert resources to that central strategic project, or to hire additional people if necessary to get it done? Who "called the shot?"

Allow me one more illustrative quote, which I found flabbergasting yet entirely consistent with the outlook I've found in lengthy discussions with all of the executives.

> "I've been planning in-house support for the last three years.... We have actually initiated the first in-house support contract with a major account, but we're still wrestling with the actual format for it. I think John's memo has probably accelerated that and maybe reduced some of the opposition to it in Autodesk. He's made a number of changes we wanted to make a lot easier for us." There's an attitude, Davies said, of "if John says it, he's God, therefore it must be true."[374]

These are not the words of an executive. If an executive wishes his company to provide in-house support, it does not take three years and wielding the bludgeon of a memo from a founder to implement his plan. An executive will draw up the plan, present it to his managers, and direct them to implement it to his satisfaction. If they cannot or will not, he is empowered to replace them with others who will. Japanese companies design an entire automobile from scratch in three years. And by the way, which executive was it who initiated the plan for support contracts and aggressively sold it throughout the organisation?

Autodesk's so-called executives see themselves as the prisoners of the circumstances of the company's history and current events. Autodesk needs *leaders* who understand the immense power they hold to define the company's future in any way they judge to be wise. One Autodesk executive told me, face to face, than he had been focusing on diversifying into other product lines and away from AutoCAD because he had given up on getting the software development department to produce future versions of AutoCAD which would remain competitive, and therefore growth must be sought in other areas.

Whether you agree with the diagnosis or not, the cure is not the approach of an executive. An executive defines the products which software development must develop and charges the development managers with building them. He assigns whatever resources are needed, terminating less important activities if required, and then supports his managers with consistent and vigorous leadership to aid them in completing the undertakings he has entrusted to them. If they fail consistently, then he removes them and gives the task to others who can succeed. If they also fail, he removes them and tries once again. But an executive cannot give up on the future of the product which generates essentially all of his company's sales and profits. If repeated frustration causes him to lose hope, he must not resign himself to the situation, he must resign.[375] Any other course violates his duty to the company, its shareholders, and its employees to lead them into the future.

In addition to a can-do philosophy and initiative, an executive must be *engaged* in the industry in which he operates. As a leader of an important part of that industry, he is a statesman for the industry as a whole and his company's rôle within it. He is aware of what is happening in the marketplace, what problems and opportunities developments in technology may create for the company, and is constantly seeking ways in which the company's assets can be multiplied by deploying them in ways overlooked by competitors: creating, for example, new markets for the company's existing products.

[374] Compare this attitude with Marc Stiegler's on page 648.

[375] This is an allusion to a totally bizarre conversation I had with Malcolm Davies at my house in Cornaux, Switzerland on July 2, 1991 (the office in Marin being painted, at the time, and uninhabitable due to fumes). Malcolm told me, "I decided we'd have to diversify out of AutoCAD into other products and markets because there was no way we could get the technical group to implement what we needed in AutoCAD to stay competitive." (Paraphrased from memory.)

I don't think Autodesk's executives know very much about the CAD business. Frankly I don't think they're very interested in it. Joel Orr put it like this, as best as I can recall, when I met with him recently, "When I talk to the people in Sausalito about what's going on in the industry, it's as if they're hearing these things from me for the first time." It was abundantly clear from our discussion that this situation is unique to Autodesk, at least among companies he deems successful. Joel does not encounter this ignorance within the management teams of the many other CAD and computer graphics firms with whom he meets in the course of his consulting practice.

You don't have to be an all-knowing guru to be an executive. John Sculley probably knew very little about computers or the computer business when he became president of Apple. But by the time I met him a couple of years later, he had certainly learned a Hell of a lot about it in the course of doing his job. You might not agree with his perceptions and you may consider the strategy he adopts to be a prescription for disaster, but there's no difficulty in ascertaining what his strategy is or what assumptions it's based upon. The same could certainly be said for companies like Microsoft, Borland, Intergraph, DEC, and IBM. Having a strategy doesn't guarantee success; you have to have the *right* strategy, to be sure. But having no strategy, reacting to events in a totally passive manner, almost certainly guarantees failure. Who would have predicted, or indeed could have conceived even a few years ago that Ashton-Tate would eventually be acquired by Borland? I would suggest that anybody who noted the lack of leadership and strategy at Ashton-Tate could have envisioned such an event. Companies which are vigorously growing, adding new products, and broadening the market for their existing offerings end up devouring, one way or another, others who view themselves as custodians of a single success, operating in a defensive rôle rather than as leaders.

What is Autodesk's strategy for the CAD business? Which member of the executive team has, can, or will articulate it?

To look at the future of a company, you have to see beyond current operating margins to rate of growth. Autodesk's rate of growth is falling with every passing year precisely as you'd expect when the market for its only product is maturing. And if, as Malcolm Davies said in Cadence, "We still believe the CAD business has a tremendous upside for us. We don't believe that here in the United States, the most saturated market in the world, that we're anywhere beyond 25 percent saturation of the CAD market…", then why is Autodesk's rate of growth falling, not rising? And what strategy do our executives have to reverse this situation? And which executive will develop it and lead the company to implement it?

In the almost ten years of Autodesk's history, only a single diversification has been proposed by a member of the present executive team, and that was to buy Generic,[376] another company in our own business, and was presented as largely a defensive move. Not a single diversification has succeeded. In the "World Class" awards in the current PC World, the companies below were represented by the following number of products, counting winners and runners up.

Microsoft	19
Borland	10
Lotus	10
Word Perfect	5
Symantec	5
Autodesk	3

All of Autodesk's products were in the CAD category, of course, and to some extent therefore compete with

[376] See page 489.

one another. Most of the other vendors are broadly diversified across many different areas.

It's well to remember that Microsoft was a one-product company in the late 1970s, and that Borland was a one-product company for quite a while after Autodesk was founded. Neither is today. They have outgrown Autodesk. Why? Because Bill Gates and Philippe Kahn are *engaged* in the industry, *developing strategies* to further their companies' position in the marketplace, and *providing leadership* to their organisations to carry out their plans—they are acting as executives. Ours aren't.

The reorganisation has been presented, even by the executives, as leading to a more effective Autodesk by removing the executives from positions of leadership in relation to the products. While this can, and I believe is, helping AutoCAD by replacing individuals who took a custodial approach to the product with others who will focus on the still unexploited opportunities it holds, it doesn't address the future of Autodesk as a company. If the same people who, by reorganising, acknowledged their own ineffectiveness in leading AutoCAD are still running the company as a whole, why should we expect them to do any better with it than they did with AutoCAD?

I'm not even sure they're interested. Again, Malcolm Davies told Cadence, "We really feel the whole future of computing is multimedia computing. Everything we learn in that effort, in the multimedia side, is going to directly benefit every other business we're in. If it ever becomes a giant application business in its own right, that remains to be seen, and if it does, that's going to take a long time."

Well, if it's "the whole future of computing" then it's pretty likely there's some money to be made there, somehow or other, and it would seem to me that an executive team who wants to see their company prosper in that future should be aggressively *creating* that market so when it finally arrives they end up owning it. Just like we did with AutoCAD. Instead, they'd apparently rather wait and see, and increasingly disappear from sight in a market they could have utterly owned by now, while Microsoft, IBM, and others are aggressively investing in developing multimedia and guiding its emergence in their own interests. No, instead they cut the multimedia budget because "it isn't paying off," and justify what we are spending by its benefits to AutoCAD.

Imagine if Bill Gates had viewed his plunging into operating systems based on its benefits to sales of BASIC. Or if Philippe had positioned Sidekick to pump the sales of Turbo Pascal. What is missed if you focus on AutoCAD's numbers is the number of opportunities Autodesk has missed or bungled which could have, by now, grown us far beyond our present success. There is not a single thing Borland has done which we could not have done, given the will and the competence to carry it out.

What drives me nuts about the overt symptoms of this problem, which include, for example, the "Cyberspace" issue of Scientific American where we finally coughed up the money for an ad for…Chaos(!), and the accelerating disappearance of Autodesk from consideration as a player in things we pioneered, often years ago, is that I've seen it all before. Precisely.

Around 1980, it became clear to me that I was losing it with Marinchip. In '78 and '79, I had staked out a position as a technology leader—first true 16 bitter, first 64K dynamic RAM board that worked, Unix clone operating system, Pascal compiler, etc., etc., but by the time '80 arrived almost nobody knew who Marinchip was. I didn't have the time or the money to promote and, furthermore, I didn't have the time or money for engineering to port from the dead-end 9900 chip I was using to the 8086 or 68K. I knew I *had* to do it to survive, but I just didn't have the resources.

It's a uniquely maddening kind of helplessness to see others become the "inventors" of things you were doing 2 or 3 years ago, and it hurts even more when they begin to really rake in the dough and become able to fund their consolidation of the market from cash flow while you can't even maintain your old stuff. This is what

is happening at this very moment to Autodesk in the fields of desktop video and virtual reality, and it may be beginning to occur in CAD.

What's different between Marinchip and Autodesk isn't the state of the slide, it's that Marinchip had an excuse—lack of money—that Autodesk doesn't. Most of the things that Autodesk needs to be doing but isn't can be done by telling people to do them or by writing checks. And we have lots of talented people and plenty of money.

Another telling parallel was that Marinchip's sales and profits continued to increase from 1980 until I deliberately beat it to death in 1982 in order to focus on AutoCAD. It continued to grow from momentum but it had lost the opportunity to broaden its market and ever be a real success.

What Marinchip lacked in the key period when it had its shot at success was executive leadership. I failed my company at that time, and eventually my company failed.

Although I'm the one stating the problem here, I'm hardly the first to point it out. Several years ago, Jim Stocker was telling me exactly what I'm telling you now, and not that long ago I supported his removal from the board when he continued insisting on addressing this very problem. For that matter, nine years ago Dick Elkis identified the problem precisely, but I wouldn't listen because he was talking about me. He was absolutely right. You can get incredibly lucky, but luck won't carry you as far as Microsoft has gone.

Finally, in order to succeed in implementing the strategies he develops, an executive must lead. In order to lead, one must have the confidence of the people who will follow. Today, among the employees of Autodesk who think about such things, I doubt you could find five people who have confidence in Autodesk's executives or who could begin to describe what strategy they have for the company's future. I can not name one single person who has that confidence.

Engagement, strategy, and leadership. This is what an executive must provide his company. It is what a company must demand of its executives. Autodesk's executives are managing our company but they are not leading it. Autodesk, and everybody associated with it, is suffering the consequences.

That's the problem.

Al Green Retires

On October 10, 1991, Al Green decided, as I had a little less than five years before, that there was a lot more to life than being president of Autodesk, and that the time had come to start living it by passing the torch to the next victim. Here's the press release announcing Al's intention to retire and my comment upon Al's accomplishments, issued shortly thereafter.

AUTODESK CEO PLANS TO RETIRE WITHIN YEAR
Will Remain Active as Board Chairman

October 10, 1991
For Release at 1:15 p.m. PDT

SAUSALITO, Calif.—Autodesk, Inc. announced today that president and CEO Alvar Green will retire once a comprehensive search for his replacement is conducted and an orderly transition is concluded. Green will remain active as chairman of the Autodesk Board of Directors. He anticipates the search process will be completed by the first half of 1992.

"Since 1984, I have overseen Autodesk's growth from less than $10 million in annual revenues to more than $238 million," said Green. "This year, I have directed the company's reorganization to a business unit structure that brings us closer to our customers. Now that Autodesk is structured to address the challenges of the next decade with general managers in charge of each business unit, I feel confident in passing executive leadership to a new CEO."

Green joined Autodesk in 1984 as chief financial officer. One year later, he was promoted to vice president of finance. In November 1986 he succeeded John Walker, an Autodesk founder, as president and chief executive. Green was elected chairman of the board in June 1988.

"Autodesk's success under Al Green's leadership has more than justified my confidence in him as my successor," said Walker, a programmer at Autodesk's European Technical Center. "Al's decision to remain chairman will ensure Autodesk's stability during the transition," he added.

Under Green's leadership, Autodesk became the world's fifth-largest PC-application software company with a 70% share of the worldwide MS-DOS market for computer-aided design (CAD) software. Autodesk successfully completed three public stock offerings, introduced a family of award-winning products, acquired two companies, invested in several others, and established subsidiaries in many markets including Europe, Asia, and the Soviet Union. Recipient of numerous honors, Green counts among his most significant achievements the promotion

of a participatory corporate environment that inspires creativity and productivity within the company's work force.

Autodesk, Inc. develops, markets and supports a family of computer-aided design, engineering and multimedia software products for computers and workstations. Autodesk shares are traded in the United States on the NASDAQ National Market System under the symbol ACAD™. For more information please call 415/331-0356 or GO ADESK on CompuServe(R).

– 30 –

ACAD is a trademark of Autodesk, Inc.
CompuServe is a registered trademark of H & R Block.

Date: Thu, 10 Oct 91 21:34:36 +0100
From: John Walker
To: Autodesk employees worldwide
Subject: AL GREEN'S PLANS TO RETIRE

Whenever there's a change in senior management, it's only natural to ask, "What next?", "What does it mean for the company?", and "How will it affect me?".

Choosing a new CEO is the most profound change a company can make. It's the CEO who ultimately determines the company's overall direction and strategy. History has shown that the CEO's performance, more than any other factor, determines whether a company will ultimately prosper or fail.

Changes in CEO don't happen very often—in Autodesk's near decade of history, just twice: first when I escaped in November 1986 and Al took over the company, and the second time today with the announcement of Al's decision to retire.

Finding a successor to a CEO who's stepping down at the pinnacle of success forces a company to decide, at the most fundamental level, what kind of company it is, what strategies it will pursue, and what style and substance of leadership it needs in an industry and market growing ever more complicated and competitive.

There will be ample time and opportunities to discuss all of these matters in the days and weeks before us. Today I'd like to concentrate on something else entirely.

Five years ago Al Green assumed the leadership of a raw-edged start-up company which had come so far so fast it had outstripped the ability of its founders to manage even day to day operations, no less its rapid anticipated growth. Sales were running at $50 million per year, having almost doubled in the preceding twelvemonth. All US-based employees fit easily in parts of the two Marinship buildings. It was clear to everybody, especially to me, that we were careening from crisis to crisis. Unless we started to manage Autodesk in a much more professional manner, before long we'd be blindsided by some disaster or other we didn't avert in time, especially when faced with the legal and financial obstacle course called being a public company.

On November 5, 1986, I ceased to be CEO and Al took the job. The task before him was as easy to describe as it was difficult to accomplish: "Fix it". His success has been complete and unqualified. Under his leadership,

Autodesk has grown by a factor of *five*, into a company with divisions around the world, well equipped to sustain and accelerate its rate of growth. We have entered a broad variety of new markets with rich potential. We have shrugged off competitive assaults mounted by some of the largest companies in existence. And Autodesk has emerged as the dominant global force in computer aided design and engineering.

But you knew all that. What's easy to overlook, unless you've ever run a company, especially one as fractious and energetic as Autodesk, is all the things that *didn't* happen. I'm talking about the disasters that bring down the overwhelming majority of companies who attempt to grow from $50 million to more than $200 million in only five years. None of those calamities ever befell Autodesk. Do you know why we escaped them?

It takes a special kind of courage, insight, intelligence, and perseverance to steer a downright weird company brimming with opportunity-crazed wolverines on a chaotic rocket-ride of compounded exponential growth through a technological revolution in a perilous business amid a world that's turning inside out. That's what it took to bring Autodesk from where I left it to the enviable position we share today. That's what now enables us to choose our destiny from a rich selection of alternatives. It's why a strong, re-energised, and quickly growing Autodesk can look to the future with confidence.

That's what Al Green has contributed to Autodesk in the past five years. And today I have only one thing to say.

Thank you, Al. Well done.

Incentives, Deterrence, and Defence

In the fall of 1991, a heated debate erupted on the technical group's electronic mail system about how best to protect Release 12 from piracy, it having been determined that patching one bit *in Release 11 enabled it to run without the hardware lock. As the debate spiraled upward into "if we do this and they do that and then we do this and they respond in this manner, well, then we'll do the other thing to keep them from sneaking around it by. . . ," I thought it was high time to step back and consider just why people pay for software at all rather than stealing it, especially since many of the most highly profitable contemporary software packages contained no protection whatsoever against unauthorised use. My hidden agenda was to get people thinking about what benefits Autodesk could provide our legitimate customers which could be easily denied to pirates and which might, in the mind of a moral mugwump, tilt the balance toward buying a legal copy.*

In this letter I rolled out the idea of an "Autodesk Global Village" network of bulletin board systems to a broad audience for the first time—I'd been discussing it various people in private E-mail since for a month or so.

Software Piracy:
Incentives, Deterrence, and Defence

by John Walker — October 14, 1991

Let me start by establishing some terminology for discussing the problem and potential solutions. When faced with the possibility of something nasty being done to you by somebody else, there are basically three approaches you can take to avert the unpleasantness.

Defence consists of active measures intended to make the event you're worried about impossible or at least unlikely to succeed. "If you try to hold up my store, you'll have to contend with the .45 Auto I keep in the cash drawer". Or, "Go ahead, invade; we have three times the troops and twice the tanks, and we'd like the opportunity to even the score with you."

Deterrence is a strategy based on persuading the opponent that doing whatever you're trying to prevent is a really dumb idea. Defence and deterrence are usually interrelated since a convincing defence is an excellent deterrent, but sometimes deterrence can be achieved without mounting a full defence. The "massive retaliation" strategy of NATO in the 1950's is a classic example: "Cross the border and we'll bomb you back to the Stone Age." Bombers and missiles may seem expensive, but they're dirt cheap compared to the multi-million-man army it would take to actually defeat an invasion. Closer to home, consider radar traps on the highways. If you never know when you may be nailed for speeding, you're less likely to drive fast. At least that's the theory.

Incentive strategies attempt to tilt the balance your potential adversary uses to weigh alternatives. Certainly knowledge of an effective defence or belief in a credible deterrent creates incentives to think good thoughts and do good deeds, but incentives needn't be the products of negative, defensive measures. A country enmeshed in a web of trading relationships with other countries, commerce on which its own prosperity as well as that of its trading partners is based, is far less likely to ponder aggression against them since doing so would impoverish itself.

The three strategies are "nested" in the sense that defence always creates a deterrent and deterrence creates incentives, but it doesn't go the other way. Incentives needn't involve deterrence or defence at all, nor need a deterrent necessarily be based on a defence capable of defeating the adversary. In terms of cost, defence is the most expensive, deterrence less costly, and pure incentives often have a *negative* cost—if you create incentives by cooperation and mutual benefit, you generate additional wealth for both parties.

I believe there's a tendency to focus on defence and deterrence and neglect both the value and the efficacy of incentives in inducing the desired behaviour. Think about it. Fundamentally, why doesn't everybody go around holding up gas stations? Is it because all the guys at the pumps are armed? No. In practice, very few are. Is it because the cops might be lurking behind the Pepsi machine, or you're afraid the tireless minions of the law will track you down and drag you to justice? Perhaps, but not very likely. Basically, I think the reason you and I don't hold up gas stations, or at least don't do it any more, is because we'd rather live in a world where we can go to the gas station and fill up the tank in peace rather than live in something resembling an armed camp. Consider red lights. Do you really stop at red lights because you're afraid a flic is hiding behind a tree, or because you'd rather live in a world where you can drive through a green light without slowing to a creep and looking left and right for fear somebody who didn't stop at the red?

What about the guys who *do* hold up gas stations, roar through red lights, and, for that matter, invade neighbouring countries? Have they convinced themselves that the defence can't stop them? Almost always. Is it because they aren't deterred by what might happen if they try? Sure. But is that *why they do it*? I don't think so. They do it because they believe that it's worth trying; that what they stand to gain outweighs what they risk losing, all things considered. In such a situation, it's often wiser to try to work on the incentives rather than getting stuck on defence and deterrence. Consider the following alternatives a convenience store chain might adopt when faced with a problem of frequent hold-ups:

1. Arming the third shift checkout clerks. (Defence.)
2. Giving away coffee and doughnuts to the police so they stop in whenever they drive by. (Deterrence.)
3. Never keeping more than US$10 in the cash register; all other money is in a dynamite-proof safe with a time lock. (Incentive.)

I would argue that item (3), once it had seeped through the thick skulls of the Beagle Boys, would do more to reduce the incidence of stickups than any of the other means.

What does all this mean for Autodesk and our problem of software piracy?

First, let's look at the different strategies we have adopted and can adopt to the end of preventing the theft of our products.

The hardware lock is our ultimate line of *defence*. We deploy it on the front lines of piracy and its mission is as simple and clear as that of a tank—to prevent any and all potential malefactors' running extra copies of AutoCAD by main force—by making it impossible. Like most kinds of defence, the hardware lock is expensive, virtually doubling the cost of goods for every locked copy we sell and burdening us with additional

development, QA, manufacturing, and product support costs which are difficult to calculate but certainly run into the millions of dollars a year. Like the cost of maintaining an army, we justify these expenses by arguing that, however high, they're still less than the cost of risking the alternative—we'd lose even more in sales if we abolished the lock. Other copy protection schemes also constitute defence. And like defence in the modern world, there are never-ending promises of "cheap, guaranteed effective" technological fixes which never seem to pan out in practice or only trigger a costly game of technological leapfrog between the developers of defences and those attempting to circumvent them.

We exercise *deterrence* by such measures as serialisation, personalisation (in Release 11 and afterward), and by the efforts of the AutoCops and their brethren and sistren abroad in tracking down and bringing to justice those who use our product without paying for it. This is deterrence in the purest form; nothing *prevents* you installing a US-domestic version on 10 machines, or purchasing a copy of DONKEY to patch the AutoCAD executable to allow it to run without a hardware lock. But there's that lingering worry.... Might there be a little piece of code in there that, one bright Monday morning, will wipe everything on my network? Will that guy who quit and went to work for the AutoCAD dealer turn us in? What if that fella I lent my discs to lets them get away and suddenly there's half a million copies running around with my name and serial number on them? Do I really want the president of my university to have to issue a public apology after settling a lawsuit because I used 20 bootleg copies in my lab class?

I'm not saying that any of these means are ineffective, nor that there aren't ways in which we could reinforce our defence and increase the credibility of our deterrent. But keep in mind that anything we do in those veins is essentially defensive in nature and negative in effect. When I say negative, I don't mean that Autodesk and its dealers don't benefit from the reduction in piracy engendered thereby, but that no defensive or deterrent measure benefits the user in any way except in the most tenuous and indirect way conceivable—by improving the profitability of the vendor, thereby funding updates and upgrades. That's pretty abstract though, especially when you're faced with the alternative of buying a lock buster or forking over DM12,300 for a legal copy of AutoCAD R11+AME.

What are the *incentives* to own a legitimate, fully-paid copy of AutoCAD? Well, you get a nice, hardbound manual—generally of much higher quality than what you get with a bootleg copy, and certainly more complete and useful than the various "Buccaneer Books" which masquerade as "simplified user guides" but are primarily bought by people who knock off the software. You have a dealer you can go to who may be able to help you with various problems... but then you could just as well go the local user group or ask your brother-in-law. You have the right to buy updates without finding a bootleg copy of the update. And then... and then?

What could we do, if we applied the same creativity we've used in squeezing a mainframe CAD system into 640K or building a global sales and support organisation by developing our dealer channel, to increase the *incentives* for a user to own a legitimate copy of AutoCAD rather than running a pirated copy? What additional value could we provide to legitimate users which is denied, in principle, to those who make illegal copies?

I'm asking you, not trying to sell a list of my own.

Here are some examples of incentives I've stumbled across in the week or so I've been turning this issue over in my mind. These are intended to stimulate your creativity, not constrain the alternatives.

The CD-ROM

Including a CD-ROM with AutoCAD[377] and treating it as an integral part of the value added of the product, not a glorified "bonus disc," creates a large technologically-supported incentive for owning a legitimate copy. CD-ROMs are easy to copy but very expensive (for the next few years) to duplicate. Huh? What I mean is that if you're willing to buy 600 megabytes of hard disc and fill it with the contents of a CD-ROM, there's no problem doing so, but most users will opt for using the CD-ROM that comes with a legitimate copy instead. What would be on the CD-ROM? Use your imagination . . . a full hypertext indexed reference manual . . . annotated source code for model ADS applications . . . a library of a million useful symbols . . . ten thousand fonts. . . . Let your mind free run and see what you come up with.

AutoCAD Global Village

This is my current obsession. If you're interested in this, please subscribe to the "**`Global_BBS`**" mail alias. Suppose that when you purchased a legitimate copy of AutoCAD you received an access code which would allow you, simply by dialing a local phone number (our goal would be that 95% of the AutoCAD users worldwide would be able to access AGV with a local call), to access every file contributed by any other registered AutoCAD user in the world, send mail to any other user or interest group, and access the combined knowledge of more than half a million other AutoCAD users. And if you had a pirated copy? Sorry, Charlie. (I'll leave means for access control and validation to those who know more about such matters than I—but I'm sure effective solutions exist.)

Anyway, here's a genuine and unique advantage of becoming a full-fledged member of the AutoCAD community. The AutoCAD community becomes, in fact, not just an abstract concept we talk about in publicity but something tangible—an electronic link connecting every AutoCAD user who wishes to participate with every other user in the world—providing access, in time, to the collected wisdom of virtually every being engaged in design on this planet.

In the distant past when I was a kid, there were lots of people who couldn't imagine a world in which anybody could pick up a telephone, punch 12 or 15 digits, and within seconds speak to anybody else on the planet with a similar instrument. How easily we adapt to miracles! When Autodesk was founded, most people couldn't even imagine a world in which virtually any business could send mail, within seconds, to any other business on the globe. Today, not only do radio stations in Los Angeles take requests by FAX, in the tiny village in Switzerland where I live, even the *butcher* takes orders by FAX, for pickup later in the day.

And still, to many people, a world in which any user of a software package can communicate, send files, access libraries, and ask questions of any other user of that product is something "utopian," "futuristic," or something that may happen "once Xanadu ships." People, it's something that I believe is going to be an integral part of the applications that dominate the software market by the year 2000, and it's something that Autodesk can implement, simply by deciding we want to.

In about six months, more or less. Without *any* software development.[378]

[377] This was done, for the first time, in AutoCAD Release 12.

[378] A meeting was held in Sausalito in December 1991 to discuss the Autodesk Global Village idea. A decision was made to proceed. On May 19, 1992, almost precisely six months after the go-ahead, the first customer logged on to the Autodesk Global Village BBS in California. By November 1992 more than 600 customers had dialed into Autodesk's central Global Village node, and the first AGV

By doing this we can not only utterly tilt the incentives toward owning a legal copy of AutoCAD that connects you with the AutoCAD Global Village, but in favour of AutoCAD as opposed to the other CAD systems sold by less imaginative companies that encyst their users in little islands of computing rather than uniting them in the global web of design, engineering, architecture, and manufacturing.

(Please spare me mail about "our contract with CompuServe," the "support burden," the "need to wait for a market to develop" and all the other stuff. I'm talking about creating incentives that don't currently exist, opening markets that aren't currently saturated, and developing businesses that the analysts won't analyse until somebody creates them. As that great American philosophical institution, Burger King Corporation, once said, "Sometimes you gotta break the rules".)

Your Suggestion Here

Incentives! Think about it. Let's not only concentrate on "more software in the box"[379] but also on "more value for the legitimate user." Sure, let's continue to defend against piracy and deter people from thinking of it. But let's also scour our brains for ways we can deliver additional value to the good guys—our legitimate customers, while denying those benefits, inherently, to the black hats who choose not to pay.

There will always be software piracy; all efforts to reduce it must aim at reducing the incidence, not eliminating it. No vendor has been able to find a technological fix, an "Astrodome impregnable defence" against piracy. No vendor or consortium of vendors has been able to deter piracy, except at the margin, by threats of legal action. Let's see what we can do with incentives. Can we come up with things we can make our software and our company do which, together, make it obvious to the person tempted to pirate our software that they're better off, in a purely economic sense, being inside the circle with a legal copy than skulking outside with a bootleg one?

If we can, we could not only end the adversarial relationship between software vendors and users but establish a new partnership between vendors and customers. Autodesk could, by doing this, define a new standard of "customer-orientation" for the software business in this decade and those that will follow.

And the bottom line? Well, every dollar that makes it there came, originally, from the pocket of a customer.

CD-ROM had been pressed and distributed to BBS operators around the globe. No software development was required to accomplish this.

[379] A reference to Information Letter 14. See page 606.

Searching for a New CEO

After Al Green announced his intention to retire as CEO, a search firm was retained to manage the process of finding a new CEO for Autodesk. The Board of Directors appointed a search committee, chaired by Jim Warren, which appealed for input, company-wide, on which criteria should govern our search. Here's my response.

Profile of a CEO
by John Walker — October 24th, 1991

This paper is an attempt to identify the characteristics Autodesk should look for in candidates to become the new CEO. As you'll see in reading it, I don't think there's any single precise picture that fits the job; there are essential prerequisites, but not a unique correct profile. I use male pronouns for concision and to avoid sounding like a politically correct Marin County weenie, not to specify the gender of candidates acceptable to me.[380]

Assumptions

My first assumption is that Autodesk is looking for an individual who will be a leader and a strong, highly-competent manager. Autodesk is not searching for a super-hero, nor are such characteristics needed to succeed as CEO, nor would we likely to find such a person if we were foolish enough to seek one.

Second, we cannot expect to find candidates who have already demonstrated success in running organisations of Autodesk's size and complexity. People who are currently CEOs of multinational S&P 500 companies with a billion dollar market cap, quarter billion revenue, rapid growth, gut busting margins, and more money than Scrooge McDuck are rarely out on the job market, and if they are it's probably an indication we don't want 'em. Whoever we pick will almost certainly be assuming the largest challenge so far in their life, and that's even more reason to focus on the fundamental capabilities that indicate the potential for growth into and in the job. If you find Bill Gates or Philippe Kahn standing in line at the unemployment office... hire them, but don't waste a lot of time looking.

[380] As was demonstrated, I trust, to everybody's satisfaction on April 14, 1992 (see page 807).

Requirements

- The CEO must be a qualified, professional manager with experience in an organisation of Autodesk's size and scope.

  Companies as large as Autodesk have to be run, to a substantial extent, by analysis and manipulation of financial aggregates. A CEO is doomed to fail if he doesn't understand how to do this. He must not only be able to see problems and solutions in the numbers, but know what numbers are needed to reveal what's really going on, and how to obtain them. The CEO need not have a financial background but must understand the kinds of financial controls, reporting, and diagnostics that are required to run a company like Autodesk and be able to put them in place, manage them as the company grows, and use them day to day to run the company. I believe it is essential that any candidate be conversant with such matters, and to be able to cite, limited only by confidentiality, the financial characteristics of the organisation he currently runs to an extent that demonstrates this ability.

- The CEO must be able to build a management team and use that team effectively to run the organisation.

  We will never find a candidate who is an expert in the CAD business, a marketing genius, a super salesman, a financial whiz, a superb motivator of people, a born leader, a compelling communicator, and a visionary with a talent for spotting new opportunities for the company, all at the same time. That would require a super-hero. Yet, when the door closes and Autodesk's senior management sits down to decide what they are going to do with our company, *every one* of those talents *must* be present in the room; otherwise an essential component of wise and informed business decision making will be absent, and the decisions will be taken in ignorance and will imperil the company.

  It is essential that a new CEO have demonstrated success in recruiting, operating, and managing a complex process with such a multidisciplinary team. It goes without saying that the candidate must have earned and retained the respect of the individuals with whom he works closely, and with subordinates anywhere within his organisation. A candidate must have demonstrated success in developing plans and strategies of his own and managing their implementation and eventual success by others.

- The CEO must be able to lead the company, developing the company's fundamental business strategy and then articulating that strategy inside and outside the organisation.

  This is so fundamental a part of the job of CEO that it cannot be delegated, not even to another member of the senior management team. Any candidate's background must convincingly demonstrate success in this essential qualification for any executive position.

- The CEO must understand the importance of excellence, and be effective in obtaining it from an organisation.

  Most initially-successful companies screw up because they've lost the will to do their best, or a belief that it's even possible. The CEO must set the standard for the company and manage the company to meet it. Sustained success in business comes not from superhuman effort or super genius, but from building and running an organisation that does extraordinary things with ordinary people. Any candidate must understand this and have consistently demonstrated such success in his career. Each candidate will have a background in one specific operating field or another: sales, marketing, finance, technology, whatever. They must personally excel in that field, and be able to persuade those with the same expertise that they do.

- The CEO must approach Autodesk as an entrepreneurial company faced with opportunities, not a static organisation faced with problems.

> The candidate must understand the nature and size of the opportunity Autodesk has and be excited about realising it and not approach the job as presiding over a ready-built money machine. I believe Autodesk has a genuine chance to grow into a business as large as IBM or Boeing. We need a CEO who's interested in making that happen.

Process

Your reaction to at this point is probably, "That's it? What about experience in high-tech, or international companies, or CAD, or in public companies. What about the size and nature of the organisations the person has run?"

I believe we must be interested in every aspect of each candidate's background, weighing them in terms of the extent they demonstrate his qualifications or lack thereof to meet the challenges that will be waiting on the desk the morning he sits down at it the first time. But as far as I'm concerned, the only requirements are the ones I listed.

I believe people with a wide variety of backgrounds could succeed as CEO of Autodesk, which is not to say the job is easy; only that being a CAD guru, a numbers whiz, or a super salesman isn't the single background that qualifies one to do it.

The most essential part of the process of selecting a CEO will be the interviews the candidate has with the operating managers of the company, the people he will, if chosen, have to rely on to implement the policies and strategy he develops, or develop and/or replace if they do not. If we select a candidate who is enthusiastically endorsed by Ruth Connolly, John Lynch, Pete O'Dell, Lew Goldklang, Richard Cuneo, Frank Balinson, Tom Mahood, Volker Kleinn, Scott Davis, and Marc LeBrun, I'm not going to lose any sleep over whether the person is likely to succeed in the job, even if I'd never met the guy. In the process of interviewing candidates, one of the most important indicators I expect to use is the individual's reaction to the various people he has met already within Autodesk. If the person has the right sense of people, he's probably going to be able to get the job done.[381]

As unlikely as it may seem that the list of people I reeled off would ever agree on *anything*, it is essential that any candidate not only pass the scrutiny of the overwhelming majority of these people, not only as somebody who's "OK", but as a person they're excited about reporting to. And if one or two individuals dissent, then it's likely their careers at Autodesk may not continue far into the term of the new CEO, as the first priority of any new CEO must be assembling a team he can work with effectively.

Although I certainly believe that choosing the next CEO is the most important decision Autodesk faces at this time, I'm not as worried about our choosing unwisely as most of the people I speak to seem to be. I believe there is a tremendous pool of wisdom in the operating and middle managers at Autodesk, the people who now feel so frustrated at not being permitted to carry out the plans that seem so obvious to them. I am confident that when they see a person for whom they could work, a person who would support them in turning Autodesk around, they will recognise that talent, endorse the person, and urge the board to select him. We're probably

[381] I was, at this time, uninformed and naïve about how a high-profile CEO search operates. In order to protect the careers of the candidates, who are all, after all, important executives in other firms already, and only one of whom will be chosen, it is necessary to maintain very high security during the search process, and this requires that the list of people within the company who get to interview the candidates be severely restricted. This places an even greater burden on the search firm, board members, and senior officers who do interview the candidates to ask the questions and obtain the answers that would result from a broader cross-section of interviewers.

going to have to interview more people than we'd like to, but I think we'll be able to conclude the search with satisfaction and confidence, not in frustration and acceptance of a "least-bad" alternative.

Conclusion

If all of this seems too amorphous, I think in part it's because we won't get a real feel for what we're looking for until we do a little looking. Just as in buying a house, the best way to understand what you're looking for is seeing several alternatives; it helps you sort the truly important from the merely desirable.

If we can agree on a set of fundamental requirements, whether these or others, whether a short list or a long one, it'll help sort out people who simply haven't demonstrated the qualifications for the job. Then it's up to us, through interviews, to decide, among a series of qualified people, who is most likely to succeed.

European Software Centre

In May of 1991 I left the United States for Switzerland, joining Kern Sibbald as the second employee of the nascent European Software Centre located in the Canton of Neuchâtel. By November the office was in operation and 7 people were already working there on various projects. The Canton's Economic Promotion office invited us to hold a Grand Opening to celebrate Autodesk's arrival in the Canton, and since our office did not have a room suitable for such an event, arranged for us to hold the ceremony in the Cantonal Parliament chamber in the Chateau.

The presentation consisted of my general talk about "Autodesk's technology and business," details of AutoCAD and how it is used, presented by Kern Sibbald, an overview by Volker Kleinn of Autodesk's European operations and how the Neuchâtel Software Centre would support them, and finally a short statement by Al Green formally opening the centre. Temporarily installing a video projector and big screen in a castle, portions of which are more than 800 years old, is challenging, but not as daunting as making your debut as a speechmaker in French before several hundred people, including many members of the government. Other than Al Green's brief remarks, everything was in French, and to double the fun, we had do the whole thing twice—once in the early afternoon so the daily press could make their deadlines for the next day, then once again at the end of the day for the actual invitees.

I went through 64 drafts of the following talk before I felt it safe enough to try on human subjects. I was afraid I'd pick up the paper the next day and see a comment on the business page like, "Autodesk founder John Walker gave a 25 minute talk about the company in the Chateau yesterday, but since he spoke in Bulgarian nobody had any idea what he said." Actually, it went pretty well, and the message seems to have gotten across. Practice didn't make perfect, but it did make the event survivable.

The English translation is my idiomatic rendering of the original French which attempts to convey the message of the talk, rather than every tiny linguistic detail of the text.

Inauguration du bureau à Neuchâtel

13 novembre 1991
par John Walker au Château de Neuchâtel

Mr. President, ladies and gentlemen:

Monsieur le Président du Conseil d'État,
Mesdames et Messieurs,

As a child, I had two passions: the dream of becoming an engineer and a love for classical music. Every day at school I studied to learn what might someday help me turn the designs of my daydreams into reality. Every night, beside the radio, I exulted in discovering the musical heritage of mankind. The host of the radio program I listened to admired, above all other conductors, Ernest Ansermet. If science and mathematics seemed to me the embodiment of truth, then *L'Orchestre de la Suisse romande* was synonymous with all that was beautiful.

Occasionally, I wondered where, or what, "Suisse romande" was. As far as I knew it might be a village in the Himalayas or an island in the South Pacific. But as the music from that distant land made my spirit soar, I thought, "certainly it must be a beautiful place."

Designing products, starting companies, and building businesses leaves little time for travel and recreation. Not until last January did I finally see Suisse romande with my own eyes. Indeed, it is a beautiful place. Never in the naïve dreams of childhood did I imagine that someday I would be an engineer and live here.

First, I would like to express my sincere thanks to the government of the Republic and Canton of Neuchâtel for the warm welcome they have extended to Autodesk. I would also like to especially thank Dr. Karl Dobler and his team for their tireless efforts in helping us open our office. We have opened many offices in many countries, but never before have we encountered such helpfulness and professionalism. You are the best in the world. Thank you all.

Quand j'étais enfant, j'avais deux passions: je rêvais de devenir ingénieur et j'adorais la musique classique. Chaque jour, à l'école je m'efforçais d'apprendre ce qui pourrait m'aider à transformer les dessins de mes rêveries en réalités. Tous les soirs, à la radio, je me réjouissait de découvrir l'héritage musical de l'humanité. Le présentateur du programme admirait par-dessus tout Ernest Ansermet. Si la science et les mathématiques m'apparaissent comme l'incarnation de la vérité, *L'Orchestre de la Suisse romande* était, pour sa part, synonyme de beauté absolue.

À cette époque, je me suis parfois demandé où était ou ce que pouvait être cette ≪Suisse romande≫. Était-ce un petit village dans l'Himalaya, ou une île de l'océan Pacifique? Une chose est sûre: chaque fois que la musique de ce pays lointain élevait mon âme, je pensais: ≪Il doit s'agir d'un pays merveilleux≫.

Entre la conception des produits, le lancement d'entreprises, et le développement de sociétés, il ne reste que peu de temps pour voyager ou se délasser. Ce n'est donc qu'en janvier dernier que j'ai découvert la Suisse romande. C'est vraiment une contrée magnifique! Mais j'étais loin, dans mon enfance, de me douter que j'y travaillerais un jour comme ingénieur. . . .

Je tiens à remercier sincèrement le gouvernement de la République et Canton de Neuchâtel pour l'accueil chaleureux qu'il nous a réservé. Je voudrais aussi exprimer ma gratitude à M. Karl Dobler et à son équipe pour les efforts qu'ils ont déployés pour nous aider à ouvrir notre bureau. Nous avons ouvert de nombreux bureaux en plusieurs pays, mais nous n'avions jamais trouvé, jusqu'ici, une telle serviabilité et un tel professionnalisme. Vous êtes les meilleurs du monde. Merci à tous.

I'd like to describe the outlines of the history of Autodesk, the nature of our company and its products, and what I see as the opportunities that Autodesk will be focusing on here in Neuchâtel in the coming years. The next two speakers, Messrs. Sibbald Kleinn, will describe Autodesk's products, our market, our financial results. I will concentrate solely on two aspects of Autodesk: technology and opportunity.

Less than 10 years ago, when I decided to create Autodesk, it was easy to concentrate on technology and opportunity. We had few distractions; we had little money, few people, no products, and no customers. But we knew that we were living though an unprecedented period in the history of technology. We knew that every 18 months, the computing power available at a constant price was doubling. This trend had been underway for years and there was no reason to believe it would slow or stop in the foreseeable future.

As entrepreneurs, we understood that this technological revolution would create enormous opportunities. Software which had, only a few years before, required large, expensive computers, being sold in small quantities at great cost to a few customers, could become, for the first time, mass market, generally-available products.

Indeed, it was possible to replace special-purpose machines with software products, sold for a fraction of the price, which ran on low-cost personal computers.

In 1982, Autodesk undertook the challenge of placing computer aided design in the hands of every engineer, every architect, every draftsman, surveyor, and designer in the world. The story of Autodesk is the story of our progress, to date, in achieving that goal.

Je voudrais rappeler brièvement les grandes lignes de l'histoire d'Autodesk, expliquer la nature de notre société et de ses produits, et vous dire ce que nous envisageons de faire à Marin[382]ces prochaines années. Messieurs Sibbald et Kleinn évoqueront ensuite nos produits, notre marché, et nos résultats financiers. Je me limiterai, quant à moi, à deux aspects de notre activité: la technologie et les innombrables champs d'application qui s'offrent à Autodesk.

Il y a moins de dix ans, lorsque j'ai décidé de créer Autodesk, il était facile de se concentrer sur la technologie et les occasions s'offrant sur le marché. Nous avions peu de distractions: peu d'argent et d'employés, pas de produits, et aucun client. Mais nous étions persuadés que nous vivions une époque sans précédent dans l'histoire de la technologie. Nous savions que tous les dix-huit mois, la puissance de calcul des ordinateurs doublait sans que le prix du matériel n'augmente. Ce phénomène, observé depuis plusieurs années déjà, n'avait aucune rasion de ralentir ou de s'arrêter.

Comme entrepreneurs, nous avons compris que cette révolution technologique créerait des possibilités commerciales sans précédent. Les logiciels qui, quelques années auparavant, ne tournait que sur des gros ordinateurs de prix élevé, vendues en petites quantités à quelques rares clients, allaient devenir, pour la première fois, des produits aisément disponibles, destinés à un large marché.

En effet, il devenait possible de remplacer certaines machines avec des logiciels vendues pour une fraction de leur prix et tournant sur les ordinateurs individuels bon marché.

En mille neuf cent quatre-vingt deux, Autodesk a relevé le défi de mettre la conception assistée par ordinateur dans les mains de chaque ingénieur, architecte, dessinateur, géomètre, et inventeur du monde. L'histoire d'Autodesk est le reflet de notre progression vers ce but.

[382]Meaning our office in the commune of Marin, in the canton of Neuchâtel, as opposed to Marin County, California!

What we do at Autodesk is so simple to understand and so easy to explain that sometimes it goes past so quickly people don't comprehend. Autodesk products build computer models of real-world objects, then manipulate them in various ways.

This simple concept, whether you call it Computer Aided Design, or Computer Aided Engineering, or Computer Integrated Manufacturing, or Molecular Modeling, or Desktop Video, or Computer Aided Drafting, is at the heart of the technological adventure of the second half of the Twentieth Century and will form the centrepiece of the industrial revolution in the Twenty-First.

The past two decades have witnessed the digitialisation of one aspect of technology after another. The word processor digitialised the typewriter. The compact disc digitalised the phonograph. The facsimile machine is digitalising the mail. Within the next few years photography, radio, and television will be digitalised.

Autodesk is engaged in digitalising the world of design, engineering, and manufacturing. To do this, we develop tools to create, inside the computer, faithful models of the objects that designers conceive.

Our products are the link between the designer's mind and the memory of the computer. Once a models is built, we can use the computer to prepare drawings for a machinist or construction worker. We can analyse the design, inside the computer, calculating weight and strength, permitting the designer to explore alternatives before the object is manufactured.

Ce que nous faisons chez Autodesk est facile à comprendre. Mais, comme nous l'expliquons parfois trop vite, les non-initiés ont de la peine à nous suivre. Les produits d'Autodesk permettent de créer, dans l'ordinateur, des modèles d'objets réels, avant de les maniupler de diverses manières.

Cette idée très simple, que vous l'appelliez *conception assistée par ordinateur*, ou *fabrication assistée par ordinateur*, ou *ingénierie assistée par ordinateur*, ou *modélisation moléculaire*, ou *logiciel vidéo*, ou *dessin assisté par ordinateur*, est au cœur de l'aventure technologique de la deuxième moitié du vingtième siècle et jouera un rôle central dans la révolution industrielle du vingt-et-unième siècle.

Pendant les vingt dernières années, nous avons assisté à la numérisation d'un domaine de la technologie après l'autre. Le traitement de texte a numérisé la machine à écrire. Le disque compact a numérisé le phonographe. Le téléfax est en train de numériser le courrier postal. D'ici quelques années, la photographie, la radio, et la télévision seront numérisées à leur tour.

Autodesk s'efforce de numériser les mondes de la conception, de l'ingénierie, et de la fabrication. Dans ce but, nous développons des outils pour créer, dans l'ordinateur, des modèles exacts d'objets.

Nos produits deviennent le lien entre le cerveau du concepteur et la mémoire de l'ordinateur. Une fois les modèles batîs, nous pouvons utiliser l'ordinateur pour préparer des dessins destinés à un mécanicien ou à un travailleur de chantier. Nous pouvons analyser l'objet, dans l'ordinateur, pour permettre à l'ingénieur de calculer son poids et sa solidité et d'explorer autres solutions avant que l'object soit fabriqué.

From the computer model we can create pictures so real they appear to be photographs. Architects can show their clients the buildings they design before construction is begun. We can animate the model, actually walking or flying through the world we've built inside the computer. The computer allows us to go where our bodies cannot go and see what our eyes do not: we can watch the valves open and close inside the combustion chamber of an engine; perch on the tail of the *Hermès* spaceplane as it re-enters the atmosphere for the first time; watch particle interactions in the accelerator at CERN; and observe the molecules designed in pharmaceutical laboratories.

D'après le modèle informatisé, on peut créer des images si réelles qu'elles paraissent être des photographies. Les architectes peuvent montrer à leurs clients les bâtiments qu'ils ont conçus bien avant que le démarrage de leur construction. On peut animer le modèle et marcher ou voler à travers le monde qui a été construit dans l'ordinateur. L'ordinateur nous permet d'aller là où nos corps ne peuvent pas se rendre, et voir ainsi l'invisible. On peut ainsi regarder les soupapes qui s'ouvre et se ferme dans la chambre de combustion d'un moteur; se percher sur la queue de l'avion spatial *Hermès* pendant sa première rentrée dans l'atmosphère; observer les interactions des particules à l'accélérateur du CERN; ou examiner les molécules conçues dans les laboratoires pharmaceutiques.

All of these things are possible today, using the products we have built in the last ten years. All of the images you've seen on the screen are digital images created from digital models that exist inside a computer, even these flags and the computed clouds in the software sky. Every image, and every model it represents was created by an Autodesk product.

Toutes ces choses sont actuellement possibles avec les produits nous avons développés pendant ces dix ans derniers. Toutes les images vous avez vues et que vous verrez à l'écran ne sont rien d'autre que des images numériques créées à partir de modèles numériques mémorisés dans un ordinateur—même les drapeaux et les nuages numériques dans le logi-ciel.[383]Chaque image, et chaque modèle qu'elle représente a été créée avec un logiciel Autodesk.[384]

When we started Autodesk, most experts said that computer aided design was not possible with a desktop computer and even if it became possible at some time in the distant future, it would forever be so complicated as to be inaccessible without lengthy, difficult, and specialised training.

Quand nous avons fondé Autodesk, la plupart des experts estimait que la conception assistée par ordinateur n'était pas possible avec les ordinateurs individuels et que, à supposer qu'elle le devienne quand meme dans un lointain avenir, elle resterait à jamais si complexe que personne ne pourrait y avoir accès sans disposer d'une formation longue, difficile, et spécialisée.

Yet today, more than half a million people around the world are using Autodesk's products to build computer models. They are designing objects ranging in scale from atoms and molecules to maps of entire continents. Our customers have shown the experts to be wrong again and again, as they have put three dimensional design, solid modeling, photorealistic image generation, and animation to use just as quickly as the tools became available to them.

Et cependant, aujourd'hui, plus d'un demi-million d'utilisateurs dans le monde entier se servent de notre logiciel pour bâtir des modèles informatisés. Ils conçoivent des objets dont l'échelle s'étend de l'atome aux cartes des continents entiers. Nos clients ne cessent de démontrer aux experts qu'ils se sont trompés, car ils ont su de servir, sitôt que de nouveaux outils devenaient disponibles, du dessin tridimensionnel, de la modélisation volumique, et de l'animation.

[383]I was *really* proud of this pun. Everybody who reviewed early drafts of the speech took it out, but I put it right back in every time.

[384]We had set up a large projection TV screen, and before the presentation we ran a series of animations, images, etc. created with various Autodesk products.

Ten years after we founded Autodesk, we see even greater opportunities ahead.

Sometimes people refer to the Industrial Revolution as something that happened in the past—an event that has come and gone, symbolised by coal mines, steel mills, and railroads. Perhaps studying history blinds us to the fact that we're living through history and, by our day to day acts, making the history that our grandchildren will learn in school. The industrial revolution began long before we were born and its end is nowhere in sight.

Each generation has bequeathed to the next knowledge, opportunities, and wealth beyond the imagination of its parents. The technological frontier of one generation becomes the starting point for the next. Electricity, telephones, radio, and aviation were once considered miracles. The centrepiece of the technological adventure of our generation is the computer. Properly applied, it will change the way every product in our world is designed and manufactured. This will, I believe, usher in the Golden Age of Engineering. Along with my colleagues, I founded Autodesk to build the tools for this age.

The computer power we need is at hand, or will arrive in the near future. The progress in the last decade is astonishing, even if you have lived through it. A task which took half an hour on a personal computer of 1982 can be completed today in less than a third of a second. If your automobile had increased in speed by a similar amount since 1982, today you would be able to drive to the Moon in about 25 minutes.

The end of this unprecedented growth in computer power is nowhere in sight; certainly not within the next decade. We can, with confidence, look at what is being done today on the largest supercomputers and expect, before long, to have the same capabilities on the computers on our desktops and in our pockets.

Dix ans après notre fondation, nous entrevoyons des possibilités encore plus grandes.

On parle parfois de la révolution industrielle comme d'une époque appartenant au passé, symbolisée par les mines de charbon, les aciéries, et les chemins de fer. L'étude d'histoire nous masque peut-être le fait que nous faisons l'histoire que nos petits-enfants apprendront à l'école. La révolution industrielle a commencé bien avant notre naissance et sa fin n'est pas pour demain.

Chaque génération a légué à la suivante des connaissances dépassant l'entendement de ses parents. Les frontières technologiques de chaque génération deviennent la ligne de départ pour la génération suivante. L'électricité, le téléphone, la radio, et l'aviation furent jadis considérés comme des miracles. L'ordinateur est l'élément clé de l'aventure technologique de notre génération. Utilisé rationellement, il modifiera la façon dont chaque produit est conçu et fabriqué. Nous entrons ainsi, j'en suis persuadé, dans l'Âge d'or de l'ingénierie. Avec mes collègues, j'ai fondé Autodesk pour créer les outils de cette époque.

Les ordinateurs dont nous avons besoin existent ou seront disponibles sous peu. Même pour les informaticiens, le progrès réalisé au cours de la dernière décennie ne cesse d'étonner. Une tâche qui prenait une demi-heure sur un ordinateur individuel de l'année mille neuf cent quatre-vingt deux peut être éxecutée, aujourd'hui, en moins d'un tiers de seconde. Si la vitesse de votre voiture avait augmenté dans la même proportion depuis dix ans, vous pourriez vous rendre sur la lune en quelque vingt-cinq minutes.

Il n'y aucune raison de croire que cette augmentation de la puissance des ordinateurs touche à sa fin. Elle se prolongera sans aucun doute jusqu'à la fin du siècle. Je suis certain que ce que nous sommes capables de faire aujourd'hui sur les plus grands superordinateurs pourra se faire, avant longtemps, sur nos ordinateurs du bureau et même sur nos appareils de poche.

Autodesk's challenge, and Autodesk's opportunity today lies in developing software products to turn the potential latent in this computing power into the tools for the next generation of designers. What we have done so far is but a small fraction of what we know we can do in the next ten years.

What convinced me of the promise of the microcomputer when I founded Autodesk was simply walking through offices and observing that every desk had a telephone but not one desk in a hundred had a computer. Knowing what computers could do, it was obvious to me that before long, whether in three, or five, or ten years, the computer would be perceived as just as essential a tool of business as the telephone.

What convinces me that the opportunity before Autodesk today is even greater than that of 1982 is observing that most architecture, engineering, and manufacturing is, even today, done in much the same manner as 50 years ago. Certainly, we use calculators and computers instead of slide rules, computer aided drafting instead of drawing boards, and numerically controlled machines instead of manually-operated ones. And yet the computer has had little true influence on the process of design and manufacturing. All of this is about to change.

One need only look at the largest aerospace and automotive companies to see the outlines of the future. The tools of the architects and engineers of tomorrow exist today in the leading schools of architecture and industrial design.

Notre défi, aujourd'hui, consiste à créer des logiciels capables de transformer la puissance potentielle de l'ordinateur en outils destinés à la prochaine génération d'ingénieurs et d'architectes. Ce que nous avons réalisé jusqu'ici n'est, dès lors, qu'une infime partie de ce que nous nous sentons capables de faire au cours des dix prochaines années.

Ce qui m'a convaincu de l'énorme potentiel du micro-ordinateur, quand j'ai fondé Autodesk, c'est que dans presque toutes les enterprises que je visitais, chacque bureau avait un téléphone, alors que seulment un bureau sur cent était équipé d'un ordinateur. Sachant ce que les ordinateurs étaient capables de faire, il était évident qu'avant trois, cinq, ou dix ans, l'ordinateur serait considéré comme un outil aussi essentiel que le téléphone.[385]

Aujourd'hui, je suis convaincu que l'avenir est encore plus prometteur. En effet, le processus de conception de l'ingénieur et de l'architecte, ainsi que la fabrication, font encore largement appel à des méthodes datant de cinquante ans. Bien sûr, nous utilisons des calculatrices et des ordinateurs au lieu des règles à calcul, le dessin assisté par ordinateur à remplacé les planches à dessin, et des machines à contrôle numérique ont pris la place de celles qu'il faillait commander manuellement. Cependant, l'ordinateur ne joue encore qu'un petit rôle dans le processus de conception et de fabrication. Cela va changer sous peu.

Si l'on observe les plus grands constructeurs d'avions et d'automobiles, on entrevoit déjà l'avenir que connaîtront égalment les petites et moyennes entreprises. Quant aux outils qu'utiliseront les architectes et ingénieurs de demain, on les trouve aujourd'hui déjà dans les grandes écoles d'architecture et de conception industrielle.

[385] See the discussion of this in Information Letter 1 on page 37.

We believe that when this next generation of computer aided design tools become available, not just to the few, but to the millions of designers around the world, when they can be run on low-cost desktop computers, that we will truly enter the Golden Age of Engineering. We are working toward that goal.

Chez Autodesk, nous estimons que lorsque cette prochaine génération d'outils de conception assistée par ordinateur deviendra disponible, non seulment à un petite nombre de créateurs mais à des millions d'entre eux dans le monde entier, et lorsque ces outils pourront tourner sur les ordinateurs individuels à prix raisonnable, nous entrerons véritablement dans l'Âge d'or de l'ingénierie. Nous travaillons avec cet objectif en vue.

We monitor the evolution of technology. We talk to our customers; they are the true pioneers of computer aided design. We monitor research, and we listen carefully to the experts, especially when they tell us what is "impossible."

Nous observons l'évolution de la technologie. Nous nous entretenons avec nos clients car ce sont eux, les vrais pionniers de la conception assistée par ordinateur. Nous nous tenons au courant de la recherche et nous écoutons attentivement les experts, surtout quand ils nous signalent ce qui leur paraît ≪impossible≫.

And then we get to work, since many hours of tedious labour separate every shining dream from the satisfaction of making it a reality.

Ensuite, nous nous mettons au travail, car de nombreuses heures de dur travail séparent le rêve génial de sa réalisation.

Which brings us back to Neuchâtel or, more precisely, to Marin.

Cela nous ramène à Neuchâtel, ou plus précisément, à Marin.

There, a small group of technologists are building Autodesk's next generation of products. Already, the Autodesk European Software Centre has contributed to Autodesk products which will be sold, starting next year, not just in the European market, but also in the United States, Japan, and all around the world.

C'est ici qu'une petite équipe de technologistes travaille à la création et au développement de la prochaine génération de produits Autodesk. Notre nouveau Centre européen de logiciel a déjà contribué à la création de nouveaux produits Autodesk qui seront vendus, l'an prochain, aussi bien aux États-Unis, au Japon, et dans l'autres pays d'outre-mer qu'en Europe même.

Already, those of us who have come to Neuchâtel feel at home here. How could it be otherwise? We are engineers and technologists; for centuries *Suisse romande* has exemplified creativity in design, craftsmanship in manufacturing, and the quest for excellence and perfection—the very ideals we strive toward. We hope, in joining you here, to continue in your tradition.

Ceux d'entre nous qui sont venus s'établir à Neuchâtel s'y sentent déjà à l'aise. Comment pourrait-il en être autrement? Nous sommes ingénieurs et technologistes. Or, pendant des siècles, la Suisse romande a donné l'exemple en matière de création, de savoir-faire manufacturier, et de recherche de la perfection—qualités que nous cherchons aussi à posséder. En nous joignant à vous, nous espérons perpétuer vos traditions.

We shall seek to build, here in Neuchâtel, now in the 1990s, the products that designers in every field, in every country, in this decade and those that follow, will use to create the world that children of today who dream of becoming engineers envision in their daydreams.

Nous tenterons de créer, ici à Neuchâtel, dans les années nonante, les outils que les créateurs de toute industrie et de tout pays utiliseront pendant cette décennie et les suivantes, pour construire le monde auquel rêvent parfois les enfants d'aujourd'hui qui songent à devenir ingénieurs.

I sometimes wonder how many people who live through what in retrospect is deemed a Golden Age realise it at the time. Did the people who first heard the works of Mozart and Beethoven I discovered in my childhood, from that distant place called “Suisse romande,” realise they were living through a golden age of music?

How many people today, among the cares of everyday life, recognise, even in passing, that ours is the Golden Age of Engineering? Will the engineers of today be remembered and revered like the sculptors and architects of Greece, the painters of the Renaissance, or the composers of the 18th and 19th centuries? Not likely, but their work will just as surely endure.

Masterworks are born of genius and realised by tools. At Autodesk, we’re content to make the tools and leave the genius to our customers—those who use them. The better the tools we build, the more people we enable to use them, the greater will be the legacy of their creativity.

We are proud to be here, and proud to join the many distinguished companies already established in Neuchâtel. Perhaps, in the next century, when historians chronicle the Golden Age of Engineering, they will consider Neuchâtel as one of the centres of that age.

Thank you again for welcoming us to Neuchâtel with such warmth!

Thank you.

Je me demande parfois si les gens qui ont la chance de vivre ce qu’on appelle après coup un Âge d’or réalisent leur chance. Ainsi, est-ce que les premiers auditeurs des œuvres de Mozart et de Beethoven que j’ai entendues dans mon enfance depuis une lointaine contrée appelée ≪Suisse romande≫, ont realisé qu’ils vivraient à l’Âge d’or de la musique?

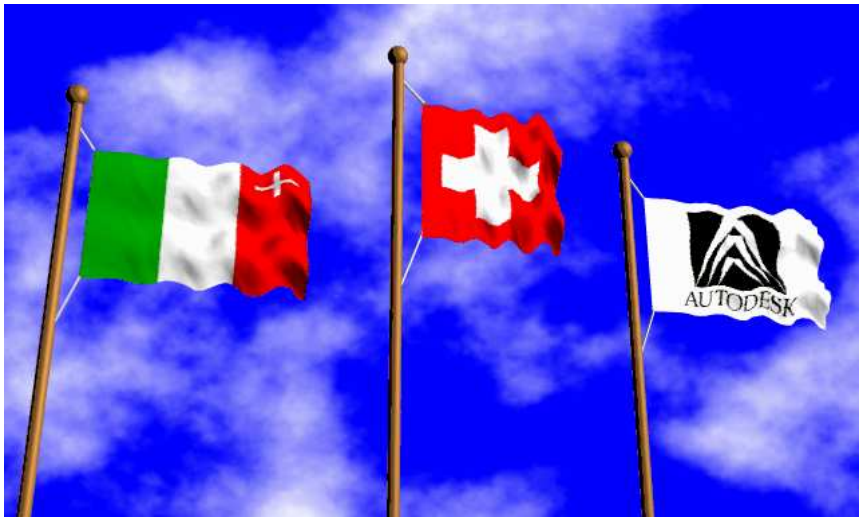
Combien d’hommes, dans la tourmente de la vie quotidienne, réalisent-ils aujourd’hui que notre Âge d’or est celui de l’ingénierie. Se rappellera-t-on un jour les ingénieurs d’aujourd’hui comme les sculpteurs et architectes de la Grèce, les peintres de la Renaissance, ou les compositeurs des dix-huitième et dix-neuvième siècles? C’est peu probable, mais leur œuvre subsistera malgré tout.

Les chefs-d’œuvre sont engendrés par la génie et realisé par des outils. Chez Autodesk, nous nous satisfaisons de créer les outils en laissant le génie à nos clients—leurs utilisateurs. Meilleurs sont nos outils et plus nombreux ceux qui pourront les utiliser, plus grand sera l’héritage de leur créativité.

Pour terminer, je tiens à dire combien nous sommes fiers de nous joindre aux nombreuses entreprises de renom déjà établies à Neuchâtel. Peut-être bien que les historiens du siècle prochain considéront ce canton l’un des centres de l’Âge d’or de l’ingénierie?

Merci encore de nous avoir accueillis à Neuchâtel avec une telle chaleur!

Je vous remercie de votre attention.



The Dark Night of the Soul

The last quarter of Autodesk's fiscal year has always been the most difficult. Ending in January, it includes Autodesk's Annual Week of Rest, when the company closes for between one and two weeks, the U.S. Thanksgiving holiday, the industry-wide stand-down around the Fall COMDEX show, and the impact on near-term sales of products announced at COMDEX but not yet shippable. In January of 1991, Autodesk ended its six-year string of consecutive quarters with rising sales and earnings by reporting earnings over two million dollars less than the preceding quarter, and only slightly more than the corresponding quarter the previous year. On January 25, 1991, the date of the announcement, the stock dropped from 52 to $40\frac{1}{4}$—shedding more than 22% of its value in a single day. There were plenty of reasons, however, to consider this an aberration—this quarter included all the war-nerves attendant to the "Desert Shield" military build-up following the invasion of Kuwait, and the eruption of a full-scale shooting war, "Desert Storm," with the U.S. bombing Baghdad a mere week before Wall Street bombed ACAD. Autodesk stock quickly recovered its losses and, by March 7, closed at 53.

The next three quarters resumed the growth in sales and earnings, making the January 1991 quarter seem an isolated event. Behind the numbers, however, were hidden increasingly desperate efforts to stimulate sales and, by cutting expenses in all sectors of the company, increase profit. Reducing investment in marketing, sales promotions, and new product development was eating the seed corn and guaranteed that when the final undoing came, it would be especially severe.

In December 1991, when Volker Kleinn and I visited Sausalito and won acceptance for our turn-around plan, I heard no hint that there was a problem with the January quarter. It was only after I arrived in Sausalito for my three-month stint as "Manager of Technology" that I learned, early on the Saturday morning after my arrival, that the quarter about to end was heading for a train wreck of Wagnerian proportions—not just a "flat quarter," but earnings per share plunging levels not seen in three years.

Legally, as soon as the shortfall became known within the company, we were obligated to report it publicly. There were several timid suggestions of press releases or analyst meetings in Sausalito, but I felt that if we'd screwed up so royally we should have the courage to walk into the lion's den in Manhattan and admit it in front of the institutional shareholders whom we'd impoverished to the extent of half a billion dollars or so. Only that, coupled with a concrete and realistic plan for turning around the company which signaled an end to business as usual at Autodesk, could restore enough confidence in Autodesk to make at least some of our major shareholders hang on.

The alternative was Autodesk's stock going into free-fall, which essentially guaranteed a

hostile takeover, because however poor the short-term prospects may look, a company with more than $200 million in sales, $50 million in profit, $200 million in the bank, no debt, and half a million customers becomes, at some price, attractive to somebody.

We scheduled a special shareholders' meeting in New York for January 30th, 1992, with individual meetings with key shareholders in New York and Boston on the following day. I decided that only by putting my personal reputation and the company's on the line did we stand a prayer of restoring confidence after blowing a quarter so badly, so I made a very long and very candid assessment of the company's strengths and weaknesses when it was my turn to speak. The same day the Autodesk meeting was held, George Bush and Boris Yeltsin met in Manhattan a few blocks from the Autodesk meeting. Their meeting, which proclaimed the "end of the cold war" was rather more amicable than the meeting between Autodesk's owners and their hired managers. The day after the announcement, Autodesk stock, already having plunged from its 1991 high of more than 60, dropped from $34\frac{1}{2}$ to $28\frac{3}{4}$ despite Autodesk's repurchasing more than half a million shares that day. The drop wiped out more than $140 million in shareholder net worth on that single day. The stock continued to drop, finally bottoming at $23\frac{1}{2}$ on February 19, 1992.

We'd have liked to have made the presentation to Autodesk employees first, but that would be illegal in the U.S. since we were disclosing financial results which would affect the stock. So, I scheduled an all-company meeting for Saturday, February 1, 1992 in Marin Veterans' Auditorium in which we'd repeat the presentation to the shareholders, and also have each business unit manager present their plans for the coming year. We hoped thereby to ameliorate the morale hit of the stumble and stock plunge by rolling out an ambitious, concrete plan for recovery. Despite its being a beautiful, sunny day, almost all of Autodesk's 700+ *California employees spent the afternoon in the darkened auditorium. I decided to make the theme of the meeting the tenth anniversary of the original meeting which created Autodesk—our hopes and dreams then—the current situation—and the promise that the future held for Autodesk if and when we overcame our current short-term problems.*

Remarks for the Tenth Anniversary Meeting

February 1st, 1992

by John Walker

Ten years ago, on a Saturday at the end of January 1982, I held the first meeting to organise Autodesk.

I don't recall having the time to think where the company would be or what I'd be doing ten years later.

Certainly, if I'd tried to guess, I'd never have expected I'd be in Manhattan meeting with a group of investors who have, collectively, lost more than half of a billion dollars because they put their trust in our company, trying to explain why they should continue to believe in us.

I have a pretty vivid imagination, but I'm no Franz Kafka.

This is Autodesk's dark night of the soul. What should have been a celebration of 10 years of achievement has now become instead a turning point.

Whether it is a turning point that reverses Autodesk's decline and launches our next great spurt of growth—

repeating, in the early 90s the giddy history of 1982 through 1985, or only an inflection point where slow decline accelerates into collapse, will be determined by what we do in the next 12 months—in every day of the next year that begins at this instant.

Today, as in that meeting 10 years ago, we will begin to organise ourselves for growth and success.

Our resources are immensely greater. Autodesk has never been stronger. Our sales, earnings, cash in the bank, market share, influence among customers around the globe, number of employees, channels of distribution, and breadth of products are all at all-time highs.

But we are vulnerable as never before.

We are vulnerable to a hostile takeover. If these were the takeover crazed years of the mid-80s it would have been a done deal already. And as our stock hits new lows, the stock market as a whole is fitting new highs and interest rates are at 25 year lows. Only rapid progress in re-establishing investor confidence, reflected in the price of our stock, can eliminate this peril.

We are vulnerable to competition. Our competitors know, more than ever before, that they must destroy Autodesk in order to realise their evil designs for the future of CAD. We must expect them to seize on Autodesk's "crisis" and "stumble" to attempt to take away our dealers and customers. And they know that the market we utterly dominate, DOS, is rapidly disappearing.

We are vulnerable to diversion of our management resources from vital tasks. Like it or not, management is going to have to spend a lot of time on investor relations; when you betray a bunch of people, you can't refuse to return their calls. We must expect that shareholder suits will be filed, and defending Autodesk will take further management time.[386] When stocks suffer a rapid decline, there are often inquiries into so-called "suspect trading" in the company's stock; none of this poses any real danger to the company, but it consumes additional management attention.[387] And finally, management will have to spend extra time on external communication aimed at restoring confidence in Autodesk.

It's a matter of credibility. As in '82, today Autodesk doesn't have any. We must earn it back, and we must start now.

We earn back our credibility through performance.

We attain that performance through professionalism.

Professionalism is a big thing that's the sum of little things.

It's meetings that start on time, have a clear agenda, and end on schedule.

It's decisions that, once made, *stay* made and aren't second-guessed by every person every day.

It's confidence in our mission and our strategy, avoiding the destructive carping that sows the seeds of doubt and failure.

[386] As expected, the harpie-lawyers in San Diego and Philadelphia filed a shareholder suit on February 2, 1992. Autodesk's general counsel described this action as "one of the many repugnant lawsuits filed whenever high-technology companies' stock prices drop, " asserting "the lawsuit is meritless, and (Autodesk) will defend the action vigorously." On December 9th, 1992 Autodesk settled this lawsuit by paying $5,000,000 into a settlement fund. This was described to me as a "risk avoidance business decision." Autodesk employees were provided claim forms so they could share in the bounty.

[387] Several investigations were made into employee trading of Autodesk stock around the time of the bad quarter announcement but, as far as I know, no charges were filed.

It's never passing on work to another person until it is as excellent as we can make it.

It's treating our customers as if they paid our salaries. They do.

It's regarding each commitment we make as a personal promise to deliver, a promise that will hurt other people as well as ourselves if we don't keep it.

These little things, and a host of others, add up to a big thing—professionalism. And professionalism is the way we succeed.

The metamorphosis begins here and now. Every minute is precious. Judgement is the asset we must not lose in the midst of a time of crisis.

I'd like to turn things over to Al Green now. He'll describe our recent fun-filled trip to New York and Boston, and then we'll tell you exactly what we told our Wall Street investors Thursday last.

Then we'll invite the business unit managers to explain to you how each of their business units will contribute to Autodesk's ambitious but essential goals for next year. Then Volker will describe what we must begin to do to succeed.

In that first crazy year of Autodesk, in order to succeed we had to perform, get serious, learn about the business we were in, and build credibility.

Well, here we go again.

Often, as I talk to people around here, I hear the word "tradition." "Autodesk has traditionally done this or that." Tradition is the excuse we use for failing to ask that essential question, "Why are we doing this?" Ten years ago there were no traditions, only opportunities and ideas to turn them into success.

Today there are no traditions except the ones we make from now on.

Let's get on with it.

Remarks for the Special Shareholders' Meeting

New York: January 30th, 1992

by John Walker

Precisely ten years ago at this moment, a meeting was underway in the living room of my house in California. The meeting had already run for several hours, and it would continue for several more. It would be the first of many meetings which would culminate, in April of 1982, with the foundation of Autodesk, Inc.

I had gathered the brightest, most dedicated, and most resourceful programmers I knew of in the world, because I believed that an opportunity, perhaps unique in our lifetimes, was opening before us—an opportunity to create a software company which could become, in time, one of the worldwide leaders of an industry only entering its infancy.

Today, we who now own the company that was created ten years ago through the efforts and imagination of those people and the many who joined their cause over the years, have assembled in a different place, under

different circumstances, in a very different yet strangely similar world to discuss the past, the present, and the future of our company.

To the past belong the efforts that created Autodesk's great success and the errors that lie at the heart of our current difficulties. In the present, we turn our attention to fixing the problems and remedying the omissions as we return our focus to the basic, simple principles upon which Autodesk's success has always been founded. In the future lay opportunities for Autodesk beyond even the febrile imagination of its founders a decade ago. Autodesk is uniquely positioned, I believe, among companies in the S&P 500, to grow into one of the great industrial enterprises of the twenty-first century. Seizing that opportunity, fulfilling the promise, and rewarding every investor who has placed his trust in Autodesk will require the kind of courage, imagination, and effort upon which Autodesk's early success was founded. But having ten years ago today started a company with little money, no employees, no products, and no dealers to sell them, I am confident that with the position Autodesk commands in the industry and the resources at our disposal we can, with your support, achieve the success that Autodesk has worked toward and that is due you as our investors.

At that first meeting ten years ago, and ever since, I have attempted to be as candid as possible about both the risks and the opportunities before our company. What we were trying to do was so difficult and so important that we didn't have the time to waste trying to fool one another. When I spoke to the founders of Autodesk, I told them that if they chose to join our venture, to expect the most difficult and exhausting year of their lives; sustained hard work with little immediate compensation, in pursuit of rewards beyond bounds, with no certainty of success. When I speak to the employees of Autodesk Saturday afternoon, I shall repeat those words.

I will be as candid with you today as I have been throughout the history of Autodesk. As a fellow shareholder, I share your concerns, disappointment, and anger over the current state of our company. I have devoted much of my time over the last year attempting, in various ways, to cause Autodesk to come to terms with the problems which, in my opinion, have finally brought us here today and which, now, at last are beginning to be resolved.

Although I have spoken sharply about Autodesk's strategies and Autodesk's management on several occasions, I continue to hold more than 850,000 shares of Autodesk stock, essentially all the equities I own, and I have never seriously considered selling my investment in Autodesk. Why? Because I believed, and I continue to believe, that no other company in which I could invest has as much potential for growth: short-, medium-, and long-term, as Autodesk. The problems which Autodesk must now focus on correcting stem, in my opinion, from continued neglect of the development of the business in a misguided attempt to meet short-term financial goals. These problems can be resolved only by managing the business, not the stock—making the investments required to return Autodesk to the rapid growth that it once enjoyed. High margins and upside earning surprises are the consequences of rapid growth, not the cause. Focusing on results rather than causes, valuing a penny this quarter over a market share point that will yield rising sales and earnings over the next decade, lays the seeds for the kind of dismaying performance Autodesk has reported today. As Autodesk acts to correct years of inaction and neglect, the short-term results, in terms of margins, will get worse before they get better. But as the foundation for sustained growth is laid over the next 12 months, the results will become apparent to all.

I would like to briefly explain the source of Autodesk's success as I see it and the nature of the challenge we face today. Technology does not evolve along a smooth, constant course. I use the term "technological transition" to describe those moments of discontinuous change that punctuate the long periods of steady, easily predictable development. Major technological transitions in our century have included the replacement of the vacuum tube with the transistor and the introduction of the integrated circuit. Technological transitions are times of enormous opportunity for businesses aware of the changes underway and positioned to benefit from them. Technological transitions are times of great peril to complacent leaders of industries being transformed,

unwilling or unable to change when the expectations of their customers are being reshaped.[388]

The introduction of the microprocessor-based personal computer in the mid 1970's was the technological transition that spawned the industry in which we now compete. It was clear to me when I founded Autodesk in 1982 that the personal computing industry was in the midst of a second technological transition, triggered by the advent of 16 bit microprocessors and the entry of major computer vendors, notably IBM. Almost overnight, not only did the computing power, available memory, graphics capability, and reliability of computers expand by a large factor, the expectations of the users of these machines, no longer hobbyists and experimenters, but now professionals who viewed the computer as a business tool as prosaic as a typewriter and as essential as a telephone, grew accordingly.

Leaders of the market, hardware and software, who failed to realise the changes underway in 1982, or were unwilling to make the risky and painful decisions required to adapt to the new environment rapidly disappeared as their competitors set the standards for the new era. Today we can scarcely recall their names. I founded Autodesk in 1982 because I understood the technological transition that was underway and the opportunities it was creating. I believed, and the success of Autodesk has proven me correct, that by raising the standards of the previous era, meeting the expectations of users dismissed as "unrealistic" by competitors, and expanding the envelope of applications of the desktop computer into areas conventional wisdom deemed "forever the province of the mainframe or workstation", we could succeed in creating a product which would become the world standard of its industry in that era. I was willing to fail four times, if necessary, before finding the product which was in the right place at the right time. While nothing about technological entrepreneurship is easy, if you're willing to keep trying and you have the staying power to remain in the game as it's changing during a technological transition, there's no secret to success other than hard work and responding to the needs of your customers. In other words, basic business.

Today, in 1992, we are in the midst of the most significant technological transition since the introduction of the IBM PC a decade ago. This transition is associated with the rapid adoption of Microsoft Windows as a standard application platform, but its importance transcends any one product or hardware environment. What is happening today is that all of the barriers, hardware and software, that once distinguished personal computers from engineering workstations are being erased. The ease of use, power and breadth, and fundamental production values that recently characterised expensive high-end professional computer applications are now being equaled and exceeded by mass market software sold in great volumes at a fraction of the price, usable on widely-available, affordable, industry standard computers. Whichever contender or contenders for the position of the next standard application platform prevails in the end: Windows, OS/2, Motif, Macintosh, or NeXTStep, the expectations of its users will be similar, and will far outstrip those of the users of software of the DOS generation.

In 1982, the standards of personal computer software and the price of entry into the market rose dramatically and continued to rise over the years. We accepted those standards, met them and raised them further, and rapidly established AutoCAD as the de-facto worldwide standard for computer aided design. All of our financial success over the last decade has been a consequence of this achievement. Today, AutoCAD holds a commanding market share lead in the DOS CAD market: in excess of 70%, and that lead has continued to grow over the past 12 months at the expense of our competitors.

So now what?

In the immediate future, Autodesk must and will accomplish many things, and shortly I'll give you an overview of what we're going to do this year, and when. But really only one milestone matters, and that's the one we're

[388] I explained the consequences of technological transitions more fully in the "What's Next" briefing. See page 563.

going to remain focused upon. We must, at the end of this current technological transition, emerge with the same or greater market share for CAD on the new standard platform, whatever it may be, as we currently command in the DOS market. If we achieve this goal, Autodesk's success in the next decade is assured. If we fail, nothing else will matter.

As you weigh our plans for the next year, ask yourself these questions. Do I believe that the market for computer-aided design is a short-term opportunity, or a business that will continue to grow and expand for decades to come? If computer aided design, for example, continues to grow at a compounded rate of 25% per year between now and the year 2020, what is the net present value of each market share point in that business, on the platform that emerges as the standard over those years? What is the prudent, conservative course for a business whose product is virtually synonymous with computer-aided design on the platforms of the 1980s: to invest in its business and do whatever is required to remain the standard in the 1990s, or to optimise margins at the expense of rate of growth and long-term market share?

Today, the computer aided design industry is still in its infancy. In the most basic sector of the business, two-dimensional drafting, most drawings are still done on drawing boards without a computer. The standards for three dimensional design, solid modeling, conceptual design in architecture, interactive mechanical engineering, facility planning, geographic information systems, and integrated flexible manufacturing have yet to be established. Each is a market with a potential as great or greater than our current AutoCAD business, and all are poised to grow at an accelerating rate for the foreseeable future. Autodesk is well positioned to become the leader in every one of these areas, and work is currently underway on the products which will establish our leadership.

I would like to briefly show you one of those products: AutoCAD for Windows. This is not a fuzzily-defined research and development effort from the laboratory; it will be shipped to customers on March 1st, 1992. We expect that by the end of this year, AutoCAD for Windows will have garnered overwhelming market share in the Windows-based market for professional CAD systems.

Demo AutoCAD For Windows

At this point, I vaulted from the stage to do a (carefully-scripted) demo of the in-development version of AutoCAD Release 11 for Windows, concentrating on the user interface enhancements, clipboard support, and programmability. AutoCAD Release 11 for Windows was finally announced on March 10, 1992 (see page 781).

Next, I'd like to share with you the list of products we'll be introducing in the next fiscal year, starting on February 1st. I'm calling this year "the year of the harvest," because virtually every investment we've made in technology, product acquisitions, and new business areas will result in delivery of new products to new customers in this year. In addition to the new products we'll be launching, virtually every existing product in our line will be replaced with a new release.

FY 93 Product Release Schedule

First Quarter	Second Quarter	Third Quarter	Fourth Quarter
AutoCAD R11 Windows	AutoCAD R11 RS6000	Cyberspace	Constraint Manager
AutoCAD R11 Mac	AutoCAD R12 DOS	AutoCAD R12 Windows	HyperChem Windows R2.5
AutoCAD R11 HP 700	AutoCAD R12 Sun	Animator PRO 1.5	Generic CADD Windows
AutoCAD R11 SGI Indigo	AMIX	Home Series Windows	
AutoCAD AME 2.0	AutoSketch Windows	Office Series	
3D Studio 2.0	HyperChem SGI R2		
Generic CADD 6.0	Generic 3DD Windows		
Generic Mac 2.0			
Home Series: Decks			

In addition to the launch of AutoCAD for Windows, which I've just shown you, in the first quarter we'll also be introducing AutoCAD Release 11 on the Macintosh, the Hewlett-Packard 700 Series, and the Silicon Graphics IRIS Indigo. Adding to our existing support of the Sun SPARCStation and the DECStation, AutoCAD will thus be available on a wide variety of workstation-class hardware.

In the first quarter, we'll be shipping the first major upgrade to our Advanced Modeling Extension to AutoCAD, AME 2.0. To date, we have delivered more than 60,000 copies of Autodesk's solid modeling solution, and AME 2.0 strengthens the links between two- and three-dimensional design, improves accuracy, and provides a programming interface that enables it to serve as the foundation for a series of mechanical engineering applications.

Our Multimedia business unit will ship Release 2 of 3D Studio, the award-winning three dimensional modeling, rendering, and animation tool. The previous release of 3D Studio has already captured a substantial market share in the professional video production market. Images created with 3D Studio were featured recently on the PBS television series "This Old House", and appear on the "White House—Explorations in Design" disc prepared to celebrate the 200th anniversary of the White House: this disc was used in a course taught last week by First Lady Barbara Bush. Release 2 of 3D Studio will strengthen its position through improved ease of use and image quality, while broadening the market to include preparation of renderings and animations from CAD models directly imported from AutoCAD.

Our new Scientific Modeling business unit will launch its first product, HyperChem for Windows, also in the first quarter. HyperChem for Windows brings serious, research-grade molecular modeling technology into the reach of every bench chemist in the world. Fast, powerful, and easy to use, HyperChem is poised to repeat the AutoCAD success story within the chemical, pharmaceutical, and biotechnology industries. Like AutoCAD, HyperChem stands to benefit from the ever-improving price-performance of desktop computers, and can form the core of a family of products built upon its open architecture.

Our Retail Products division in Bothell, Washington will introduce three new products in first quarter. The first, Generic CADD 6.0, actually jumped the gun—we started shipping it last week. It is a major update to the market leader in low cost two dimensional drafting, and embodies virtually every capability requested by users of previous releases. Generic CADD for the Macintosh will be updated with the shipment of Release 2.0. Finally, the next module in our ground-breaking Home Series: Decks will be shipped. In the few months since its introduction, the Home Series has created an entirely new category in the CAD market: ready-to-use, application-tailored design software for do-it-yourselfers. The four existing modules: Home, Kitchen, Bathroom, and Landscape have sold more than 30,000 copies already, and are finding shelf space in hardware stores as well as existing software outlets.

The second quarter will see the shipment of AutoCAD Release 11 on the IBM RS-6000 PowerStation. For the first time, AutoCAD will be available on the same hardware as CATIA, and will provide an entrée into large mechanical design accounts. Significantly, development of the RS-6000 version of AutoCAD was performed entirely by our new European Software Centre in Neuchâtel Switzerland, which will also manufacture the product for distribution worldwide.

The highlight for the second quarter is the first domestic customer shipment of AutoCAD Release 12. Release 12 begins by delivering what most CAD users ask for most frequently: speed. AutoCAD Release 12 is the fastest AutoCAD ever, and its drawing speed is reinforced by major ease of use enhancements including interactive editing and dialogue boxes that update AutoCAD to the age of graphical user interfaces. Incorporated in AutoCAD Release 12 are hundreds of user-requested capabilities and the customers who are now testing it seem unanimous in their evaluation of it as the most significant release of AutoCAD in years.

Think about it: if we're living in the information age, and the 1990s have begun with the global triumph of market economies, then why the devil don't we have any markets for information? In second quarter, the American Information Exchange will open for general customer trading. Already operating in a pilot phase as market makers and sellers are recruited, the American Information Exchange is a computer-mediated free market for information, expertise, and consulting services. As unlike existing online services as NASDAQ is next to a fruit stand, the Information Exchange has been hailed by industry leaders including Mitch Kapor, Portia Isaacson, and Esther Dyson as one of the most significant innovations of the decade. An information market is as central to the information age as currency futures are to international trade. We are opening the world's first information market this year, and you're all welcome to try it for yourselves.

Our Windows presence will be augmented by the second quarter release of AutoSketch for Windows, which incorporates innovative user interface capabilities which make it, perhaps, the easiest-to-use precision drawing tool ever.

HyperChem for the Silicon Graphics IRIS will be shipped in second quarter, providing a high performance migration path, totally compatible with HyperChem for Windows, that spans the performance spectrum from the affordable Indigo to machines in the supercomputer class.

The second quarter will also see shipment of Release 2 of our Three Dimensional Drafting product, a low cost tool for conceptual design and modeling in 3D.

Quietly, our virtual reality project has moved from research to development to product, and in the third quarter our efforts in that arena will culminate in the shipment of Autodesk Cyberspace, which allows third party developers to create virtual environments and interact with them. In helping our customers apply virtual reality technology to their own areas of application and incorporating it within our own products, we'll be setting the standards for three-dimensional user interfaces. As the tumult in the market for two dimensional graphical user interfaces settles down and clear standards emerge, the contest to define standards for 3D is just beginning. We believe that this technology will be absolutely central to the next generation of computer aided design systems, and we intend to be the leader in the field.

Third quarter will also see shipment of local language versions of AutoCAD Release 12 in Europe and Asia. In addition our Multimedia group will launch Release 1.5 of Animator Pro, which includes a new capability called "Lights, Camera, Action" that makes animation more accessible to the novice user.

Versions of the Home Series products will be released for Microsoft Windows in third quarter, along with the

introduction of its complement: the Office Series.[389]

In fourth quarter, AutoCAD will enter the field of parametric design with the delivery of the AutoCAD Constraint Manager. Based on a constraint solution architecture we believe to be the best in the industry, it will integrate constraint-based parametric design directly into AutoCAD, making it accessible to any AutoCAD user. This product will work with existing versions of AutoCAD Release 12 and will not require an update of AutoCAD.[390]

We'll ship the first update of HyperChem for Windows in fourth quarter, and we'll continue the migration of our products to Windows with the roll-out of Generic CADD for Windows. Thus, by this time next year, all of our CAD products from the Home Series to AutoCAD will be available on Windows.[391]

With so many new and updated products coming to market next year, and faced with Autodesk's reputation of failing to broaden its focus beyond a single product, AutoCAD, it is necessary to discuss what has changed: why Autodesk will succeed this time when it has failed so many times before. I've frequently said that Autodesk doesn't launch new products, it jettisons them, pushing them off the loading dock without the promotion and support required to give them a chance in the marketplace. Despite this neglect of products developed at great expense, few of Autodesk's products have actually ever flopped. Each of our multimedia products: Autodesk Animator, Animator Pro, and 3D Studio have the largest market share in their categories on the PC hardware. The Advanced Modeling Extension for AutoCAD is the most widely used solid modeler in the world. Between Generic CADD and AutoSketch, Autodesk leads the market for low cost drawing tools worldwide. Our Science Series products, CA Lab and Chaos, both introduced with essentially no budget for retail promotion, have actually doubled their sales forecasts for this year. Autodesk products have won award after award, and have been favourably reviewed in publications around the globe and over the years.

There's no column in the financial statement labeled "missed opportunities", yet it's only opportunities foregone out of a misguided short-term focus and unwillingness to invest in developing a market as we did in the early days of AutoCAD, which separate Autodesk from widely-diversified software companies such as Microsoft and Borland. Fortunately, the opportunities that Autodesk has missed in the past remain open: they have not yet been seized by competitors. As Autodesk moves to exploit them, we shall prove by example that Autodesk has changed.

See for yourself. With so many new products and with their vital strategic importance to our company, if Autodesk continues to bungle new product launches, it'll be obvious to everybody before long. In addition to aggressively launching our new products and updates to existing products, we have substantial opportunities to broaden the market for our existing products, especially AutoCAD, as we make up for our silence of the past few years. Release 11 of AutoCAD, the version we're currently shipping, is a comprehensive three dimensional design tool that includes solid modeling and supports a host of industry-specific applications, yet in the minds of potential customers it is often perceived as a "2D drafting package". As we communicate what we're doing, we'll also be selling what we have, and we'll grow the market for AutoCAD beyond the drafting shop.

Is the market for CAD saturated? Well, what do you mean by CAD? Even if you limit CAD to professional, two dimensional, production drafting the market is far from saturation. Every shop with one or two copies of AutoCAD and five or six drafters still on the board represents future revenue for Autodesk. Every company with dozens of drafters and no CAD system is an opportunity that awaits Autodesk's effective communication of the productivity benefits of CAD. In the larger world of CAD, where the "D" stands for "Design", Autodesk has yet to scratch the surface. With the launch of our upgraded solid modeler and constraint design package

[389] Released on schedule; terminated in November 1993.
[390] This one slipped.
[391] The Home Series was never ported to Windows.

this year, we'll begin to deliver the products we need to address this market. As we communicate the benefits to our customers and begin to accumulate success stories in the real world, the momentum will begin to build.

Autodesk doesn't need to be "turned around". It's already pointed in the right direction; it just needs to get moving again. Over the past two years, in Europe, we've made the kinds of investments in Autodesk's future that are needed now in the rest of the world. This quarter, Europe met its forecast sales, and we anticipate continued strong growth throughout the next year. Europe is a prototype for the changes we are already putting into place in the Americas and the Asia/Pacific region. Our success in Europe demonstrates the soundness of that strategy and the benefits that flow from its patient and professional execution.

Returning Autodesk to rapid growth and high profitability doesn't require any arcane knowledge or superhuman capabilities. It's simply a matter of controlling costs, making realistic forecasts, meeting schedules, promoting aggressively, selling effectively, satisfying customers, and developing products based on the needs of the marketplace. In other words, what any competent, professionally-run business must do in order to grow and prosper.

Last December, I met with the management and directors of Autodesk and, speaking as a shareholder as well as a founder of the company and employee, I asked whether it was their goal that Autodesk remain on the course I charted for it a decade ago: as a rapidly-growing, broadly diversified, highly profitable, industry leading software company. They said yes. I took them at their word. I agreed to come and help in the difficult task that lay ahead. The market position, products—current and under development, reputation, financial strength, and people that Autodesk have today equip us for success in this endeavour. The road will be long and hard. The perils are many. But the opportunity is beyond calculation, and the reward will be worth it. We are living through a period, spanning decades, in which the entire process of design and manufacturing of every product, in every industry, in every nation, will come to incorporate the computer as an integral part. In the truest sense, the world is becoming digital. Autodesk is in the best position of any company to bring about this technological transition and to benefit from its achievement.

I believe that Autodesk today is at the threshold of a second spurt of growth fully as powerful as that which characterised our early years. If we act now, we can see our company and our investment in it grow as in those heady times. If we fail to act, we shall lose everything we've worked so hard for over this last decade.

We are already at work. Remedying the damage born of years of inaction is not the work of one week or two. Changes are already underway; results will become evident as the year progresses. I do not ask for your trust. Trust must be earned through performance, and Autodesk's recent performance merits skepticism, not confidence. I ask only for your support as we do what must be done to rescue our company, and for the time to accomplish what we are undertaking. The results will speak for themselves.

It is my pleasure now to introduce Volker Kleinn, Vice President for European Operations and Chief Operating Officer of Autodesk, who will describe the specifics of the recovery of Autodesk of which he is the chief architect.

Company Meeting: Concluding Remarks

I made this brief statement at the end of the company meeting, following all the product demonstrations and presentations by the business unit managers.

And now, as in 1982, it's over. Now, as then, we go home and think over what we've heard and how much we each want to contribute to seizing this unique opportunity we share.

The world of 2002 in which we celebrate our 20th anniversary may be filled with wonders that not one person in ten million today believes are remotely possible.[392]

Those wonders will exist if we make them so. If we succeed in this endeavour, our 20th anniversary will celebrate a triumph we can barely imagine today.

Our goals for the next year are ambitious.

They must be.

Our goals for the next decade are mindboggling. Perhaps a century hence people will look back at a time called The Golden Age of Enginering and think, wistfully, “to have been there, then.”

Let it begin.

[392] A veiled reference to nanotechnology.

Reporter at Work

Imagine what it would be like to go through life believing that others are motivated only by ego and greed, that people lie whenever possible, that personal achievement and a fulfilling career can be built only upon the ruined reputations and destroyed dreams of others, and that threats, bullying, innuendo, and falsehoods are the raw material of success.

In early 1992, participating in Autodesk's management, it was easy to feel like a punching bag. In rapid succession, we had the bad quarter, the stock price collapse, the New York shareholders' meeting (see page 718), a shareholder suit, the SEC sniffing around for evidence of insider trading, a Release 12 project in trouble, and a CEO search that seemed to be progressing at a glacial pace. Then something bad happened. On January 31, 1992, I got a message saying that Greg Zachary of The Wall Street Journal would like to interview me as part of a "profile on Autodesk." Zachary was referred to our P.R. firm, and told them that he had an extremely negative story already written, and that unless he was provided access to Autodesk's senior management and an exclusive interview with me, was ready to run it. The P.R. firm characterised Zachary as "known for 'ripping companies to shreds'", and provided samples of his recent work. Then I remembered. This was the guy who splattered Jaron Lanier of VPL, a pioneer in virtual reality, the one who ran the piece on bogus "CD rot" on the Technology page, the author of the relentlessly negative coverage of Microsoft over the years in which it became one of the most successful companies in America. On one occasion when Bill Gates was giving a speech, Zachary, in the audience, shouted out "that's a lie" repeatedly.

I obtained copies of most what he'd written in the last two years, and couldn't find a single story I'd characterise as positive. Another Bay Area journalist told me that, privately, some of Zachary's colleagues at the Palo Alto bureau of the Journal considered him an embarrassment, but since his tabloid-style writing sold papers, Dow Jones headquarters loved him. Basically, Zachary delivered the following threat to Autodesk—throw him Walker, or else he'd clobber Autodesk with his "already written" story. My first inclination was to tell him to, well, you know. My second and third inclinations were the same. Zachary's threat was genuine. In early February, Autodesk stock was in the mid-20s, a level not seen since 1988, and a further drop, spurred by an unrelentingly negative story delivered to two million readers, including essentially all of our shareholders, could easily move Autodesk stock into the territory where a takeover bid would finance itself from Autodesk's cash reserves and the ongoing revenue from AutoCAD.

So, we decided to do an interview with Zachary. With a looming "copy deadline of February 14," we scheduled the interview for February 10, 1992. I video taped the interview in order to document any divergences between what we said and the interpretation in the final article. After the article appeared, I produced a video of the unedited interview with a brief

introduction. What follows is a transcript of that video. (Multimedia being, as of this writing, insufficiently advanced to permit me to embed two hours of video in this book.) At the end of the interview, on page 776, I quote a few of the many egregious misstatements and body-slams that characterised the piece which finally ran on May 28, 1992. It's astonishing how little of the content of the two-hour interview he deemed worthy to include in the final story. Another journalist who viewed the video was amazed at how ill-prepared Zachary was—he didn't know that I had left the board in 1988, hadn't looked at most of the company's products, was unaware of the current release of AutoCAD, had never heard of the European Software Centre, etc.—all things prominently mentioned in recent press releases and Annual Reports, and highly relevant to the topics raised in the interview.

If I had it to do again, I wouldn't. I'd tell Zachary to publish and be Damned, and rely upon the wisdom of our investors to weigh his pronouncements upon Autodesk in the context of his other reportage. Besides, it's hard to imagine how he could have written anything more negative than the final "profile." Seeing yourself turned into a caricature, the work of 10 years dismissed, the company you've helped build lampooned, takes a lot of wind out of your sails, especially when this is the first notice taken of the company after ten years of unbroken and record-setting success. Perhaps some day the anguish of Zachary's victims will be visited upon him, but I doubt it. I think he feeds upon it.

Title

The following text scrolls by, accompanied on the sound track by Monty Python's "All Things Dull and Ugly."

G. Pascal Zachary
of
The Wall Street Journal
interviews
John Walker
and the management of
Autodesk, Inc.

Monday, February 10, 1992

On January 30, 1992, at a special shareholders' meeting in New York, Autodesk announced its sales and earnings for the final quarter of the fiscal year.[393]

Sales for the quarter were $66.5 million, earnings $11.9 million.

Full-year sales were $274 million with earnings of $58 million.

These results were deemed "disappointing."

Autodesk stock, which traded above $60 in June of 1991, closed at 28 1/4 the next day.

[393] See page 718.

After ten years of success, Autodesk was newsworthy.

On May 28, 1992, Autodesk was called “A Strangely Run Firm” based on “A Most Unusual Interview.”

This is that interview.

Introduction

Cut to John Walker facing camera, in front of Sun workstation. In the background is a HyperChem box.

Hi, I’m John Walker.

I believe that in many ways, print is the most dangerous of media.

They call ’em “media” because they “mediate,” after all. And while all kinds of trickery are possible in video, at least you get some sense of what really happened. In the press, all the information you receive has been assembled by a reporter and presented to you in a tidy package. Most times you don’t have any way to determine whether the story represents reality or something else again. You might read, in *The Wall Street Journal*, for example, that I moved to Neuchâtel Switzerland in order to, quote, “find more seclusion.” The fact that I am one of the more than *fifty* Autodesk people who have come to Neuchâtel to work at our new European Software Centre here was left unsaid. Kinda changes the meaning, doesn’t it?

I’ve done dozens of interviews over the years with a variety of publications ranging from local newspapers to *Business Week*, but I’ve never experienced anything like this, before or since.

I believe in thorough preparation for an interview. I always obtain copies of a reporter’s recent articles, and I read them over to try to understand the direction the interview may take. I don’t do this to slant my answers toward the reporter’s prejudices, but instead to make a special effort to explain things the interviewer might have trouble with.

When you sit down across the table from a reporter and the door closes, your company’s reputation, your personal reputation, and potentially the rest of your career are on the line. If you haven’t done your homework, you won’t know whether you’re dealing with a prince of journalistic integrity or some little pencil-pushing thug bent on assassinating your character and destroying your company.

As I prepared for the interview you’re about to see, I realised it would be a unique experience; the events in the company’s recent history and the information I gleaned from studying the reporter’s work made that very clear. And rarely are the stakes higher than a Page One profile in *The Wall Street Journal* when a company is perceived to be on the ropes, searching for a new chief executive officer, and at risk of a hostile takeover if its stock should fall much further.

What goes on in an interview like that, anyway? Wouldn’t it be really *cool* to be able to eavesdrop on a session like that? I sure wished I could watch a video of some poor sucker on the hotseat to better prepare for what I was about to go through.

Not a bad idea, I thought.

And hence, this video. I set up the camera so what you see is what I saw, the reporter firing away at me. The

whole senior management of Autodesk was present also, but as you'll see, they got relatively few questions even though they'd been the people running the company since I retired as president in 1986. I'm not terribly proud of my performance in this interview. I don't like being argumentative, and every other interview I've done has been entirely cordial. But you have to be careful not to let a reporter put words in your mouth which can then be quoted back in print, and you mustn't let the reporter's questions incorrectly define the reality in your company. Also, I don't believe in letting bullies get away with it. I learned that in the sandbox, not in engineering school.

You might find it interesting to compare the picture of Autodesk you see here with the impressions you get from reading *The Wall Street Journal* piece from May 28, 1992. If you're interested in what's really going on at Autodesk, I recommend an interview I did about a month later with Mary Eisenhart of MicroTimes magazine, which appeared in the May 11th, 1992 issue.[394]

OK, so here we go. This isn't pretty, and you may decide that newspaper articles, like sausages, are best enjoyed if you don't know how they're made. As unpleasant as it may be to watch this tape, actually doing the interview was much further down the old cosmic fun scale.

Engage.

Interview transcript

Dramatis Personæ

CA	Carolyn Aver	Autodesk VP, CFO
RC	Ruth Connolly	Autodesk AutoCAD General Manager
MD	Malcolm Davies	Autodesk Executive VP
LG	Lisa Goldman	Autodesk PR firm representative
AG	Al Green	Autodesk Chairman & CEO
SM	Sandra Marin	Autodesk VP, General Counsel
JW	John Walker	Autodesk Manager of Technology
GZ	G. Pascal Zachary	*Wall Street Journal* Staff Reporter

GZ: . . . got stuck though and I got a little delayed. . . but, uhhh. . . . Well, listen I really appreciate. . . .

JW: OK, excuse me, do we have the tape started?

LG: Yes.

JW: We're video taping this, as agreed.

GZ: *(Interrupting).* Yeah, I know—that's OK.

JW: I just want to make clear that this recording is copyright 1992 Autodesk Inc. All rights reserved.[395] All participants in this on-record interview acknowledge that it is being recorded and that this recording or any derivative therefrom, including but not limited to transcripts, quotations, or edited extracts may be

[394] A transcript of *that* interview appears on page 784.

[395] The copyright was subsequently assigned by Autodesk to me, and I then placed the video in the public domain.

used in any matter by Autodesk Incorporated. No participant in this interview is a public figure.[396]

GZ: OK—that's fine. I'll just remind 'ya that this is a newspaper interview and that we'll just treat it as such. Ah, well listen, I appreciate the chance to come by and talk with y'all cause, let's see, Autodesk is a very interesting company and, uh, you know, for reasons that have to do with the fact that, uh, they're just so many things we can get to... we haven't—haven't really paid it a lot of, much attention in the couple of years I've been covering software—I also cover personal computers—but, uh, I've had a chance to do some—a little bit of boning up on the company and, you know, I'm real grateful that I had a chance to come by and talk because, uh, there's really, is no substitute for talking to senior management when it comes to assessing how a company is doing or what it's up to so, uh, I hope that... now I don't know what kind of time frame you had, but I hope we can get through a lot of things and, I, I hope also you'll appreciate that, uh, you know, a lot more about the company than I do and I will benefit by whatever time you can, you can share with me because, ah, naturally I have a lot of qu... you know... there's a lot of things I'm trying to come to grips with and... and in a lot of ways you're the best people to help me.

JW: We have until five.

GZ: Okay, allright, great, great. Ahhh... is there anything you guys wanted to mention in kicking it off, or, or, no, cause, cause otherwise I can just explain that in a lot of ways, uh, this is sort of straightforward, uh, we're—we're interested in profiling the company partly because it's, it's at a crossroads, partly because it's in tr..., you know there's some interesting issues around uh, uh, the the personalities and the, the.

JW: I thought we weren't here to discuss personalities. That's what you said.

GZ: Yeah, yeah, right. We're not going to discuss personalities. That's right. But the personalities.... People run this company, right?

JW: People run every company.

GZ: Right, and so when you write about companies, you can't avoid writing about people, and so there are some interesting folks here and the technology that you guys have... you guys have always, a strong reputation for technical excellence in the software business in particular, uh, I think that, you know, even, even in, even, even the investment community is aware that that's very important, and, and, and so, uh, uh, we'll be able to talk a little about that too. Ah, I wanted to, I wanted to, uhhh, I'm interested in talk, in talking to you all, and I hope that for some of 'ya if there are some question, that I can't pose to you within the time that we have or I just happen to forget to ask 'em, uh, which is my main nemesis in these interviews is, uh, is forgetting things, uh, that we can... maybe I can get back to you. I mean, Al, I've talked to you on the phone a couple times, but, but, in particular, John, because since this is my first opportunity to talk with you, and, uh, I was going to direct some initial questions to, to, you. Uh, so, I hope that, that, that's OK, so, basically, ah, I wanted to, I wanted to talk about, ah, the, ah, some of the management, some of the issues about the company that you've raised, and, and I actually wanted to get to 'em, get to some of them specifically, because, I th... uh, uh, a little bit down the road, because I think we can get at some of these questions of the company's future and the strategies being put into place, and likewise. But I want to open with a few general questions about how, uh, you start..., why you chose to express your misgivings about, uh, uh, Autodesk management in a, this sorta semi-public fashion, and, and, what you felt the, uh, the ramifications would be from that. I know there's a passing reference in, in the letter... in, in, the, the letter that, that you were aware that, that you had, uh, thought

[396]This is to rule out a "public figure" defence should a lawsuit for libel or slander ensue. U.S. law allows the publication of totally fabricated material if the subject is a "public figure." For regular people, publishing known falsehoods is grounds for damages.

about the consequences of, of distributing it, but why didn't you just sort of sit down with 'em, with, with people face to face, and...

JW: *(Interrupting).* Well, first I disagree with your characterisation of Information Letter 14 as speaking of misgivings. You've read Information Letter 14, I trust?

GZ: Uh, huh. Okay. We'll rephrase it to, to just say "an expression of your thoughts." Why didn't you choose to express your thoughts to management in private, face to face...

JW: I have, on numerous occasions, as I continue to.[397] Information Letter 14 is—you will note the number 14 appears in it...

GZ: *(Interrupting).* I know they're thirteen others but...

JW: *(Interrupting).* Twelve others.

GZ: Twelve others...that's fine. The question I've got though is that, you know, you, you in distributing something like this to a wider audience you naturally, uh, uh, as you yourself mentioned, copies were going to get out to people...

JW: *(Interrupting).* Well...

GZ: well, well, outside the family and

JW: well, would you like me to answer that question or would you like to make a speech?

GZ: Well, I'm trying to get you to stay, stay on that question rather than dissect the question.

JW: Well, no—you asked me a question in which you began by asking "your misgivings about the management of Autodesk." Information Letter 14 was not about misgivings about the management of Autodesk. I explicitly stated in the beginning that I had no desire whatsoever either to replace the current management of Autodesk or to take a rôle in it myself. I forget on which page, but it's right at the beginning.

GZ: Okay....

JW: Information Letter 14 is not about misgivings about the management of Autodesk. It is not about misgivings about Autodesk, so much as it's about, like the twelve Information Letters that preceded it, the state of the industry, what companies must do to compete in it, and it is entirely in keeping with the Information Letter I wrote ten years ago in which this company was organised.[398]

GZ: *(Interrupting).* Okay, did you think...

JW: The industry is changing, companies must adapt, and it was necessary to allow everybody in the company—not just two or three people at the top—but every manager within the company and most employees within the company, to understand the competitive environment within which our company competes is shifting. You can't shift the mindset of an entire company by talking to a group of five managers.

GZ: Do you undermine senior management by distributing a paper like this?

[397] See, for example, my 1988 memo to Al Green titled "Technological Leadership," which appears on page 430.
[398] See page 13.

JW: No.

GZ: You don't.

JW: All of the senior management who was in charge then are in charge now. We've reorganised a little bit, but...

GZ: Well, Al Green is outgoing, I mean, I mean one could say, *ad hoc*, that senior management was undermined, I mean in the sense that, that...

JW: You could say that. It's not true.

GZ: Well, all right, again, I mean, I mean, ah, you know the, the question really is, whatever your intentions, I mean, wasn't it inevitable that if you distribute pretty harsh criticisms of senior management like that, that it's going to undermine them, it's going to make it difficult for them to function...

JW: Information Letter 14 does not contain what I consider to be harsh criticisms of management. It contains specific questions about Autodesk's strategy—a strategy which was largely coherent with the time that I was managing the company. Information Letter 14 is a letter about the fact that the company's strategy must change, that the strategy that worked when I was president in '83, '84, and '85, and when Al was president in the subsequent years is not the best strategy for Autodesk entering the 1990's.

GZ: Okay. I mean, uh, I mean, while you didn't personally single out individual managers, I mean, you did say, on page 30, "I was so appalled at what I heard at one management meeting that I vowed never to attend another management meeting, and I never have." I, I mean, that's taking somebody to the woodshed by, by any measure if, if, you're, you're so, you're so frustrated by, uh, you know, "what disturbed me so much about this incident was the way management seemed to be taking their marching orders from the accounting rules rather than the real world."

JW: Well, well, you see...

GZ: *(Interrupting).* That's a criticism. That's expressing misgiving about management.

JW: That is a criticism about how United States public companies are run. Autodesk is a United States public company. I think you will find that most of the company you talk to—very few which have been encouraged to go out and report disappointing short-term results to make the investments that are necessary to adapt to a changing market. Autodesk, yes Autodesk had that problem. Autodesk does not have that problem any more. Information Letter 14 was the only way that I knew of to cause Autodesk to adopt a long term survival strategy for this company. We could continue to run margins of fifty percent or sixty percent for one year or two years, but we intend to be in business for the next ten, twenty, thirty...fifty years, and it was necessary to put the issues squarely in front of everybody. You can't go out and say, "We have a secret plan for the evolution of the market." You can't go the shareholders with that. You have to say, "The market has changed; the competitive environment has changed; the opportunities are unbounded, and as the management of this company we are going to make the investments that are necessary to exploit that opportunity."

GZ: Okay. Frankly, I, uh, you know, admire, a lot of the candour, and, you have, as with with regards to, the, the, climate in which companies operate under, particularly with the public equity market and the pressure that brings, okay, but.... This, really.... *(Turning to Al Green.)* What's, what's your professional training?

AG: My background is finance.

GZ: Are you an accountant?

AG: No, I'm not.

GZ: No, okay, you've got uh, what, an MBA in finance...

AG: Something like that.

GZ: Yeah, because, I mean, cause, cause one of the things, I mean, one of my questions was, was if you preside over the installation of a CEO and chairman whose background is in finance, is in, in you know, whose skills are in manipulating financial assets, ah, why would is surprise you that you get a company that starts acting like...

JW: Al Green is the best manager I have ever met and I endorsed his installation, as you choose to term it, as president of Autodesk, because I had never met anybody in my life before Al Green who I thought could run this company during a period of rapid growth. It was all I could do to run this company from zero to 50 million dollars a year. Al Green ran this company from 50 million dollars a year to more than a quarter of a billion. His record of success quite adequate as far as I'm concerned, and much better than I was expecting, I must say, at the point he took the company over. The company has continued to.... When I talk about "taking their marching rules from accountants," I'm not talking about Al Green; I'm talking about the analysts on Wall Street, I'm talking about people that are looking at...

GZ: *(Interrupting).* No, you, you, you, but you say the way management, and you're clearly referring to Autodesk management, seem to be taking their marching orders from the accounting rules rather than the real world. I don't want to get hung up on whether or not you were criticising your own management.

JW: I don't understand why that's an important issue to you, whether I was criticising management or not. I was criticising strategy.

GZ: No, the issue... yeah, cause the important issue to me... you're right—that wasn't, but the important issue is, does public criticism like that undermine your, your management's ability to lead and you said that there was no other way to get these issues on the table for people to grapple with. And that may be, I mean, you, you lived through it, I didn't, so I mean, I mean, I'm not dismissing that, I'm just saying that what happened before you reached that point in terms of the discussions you presumably did have that, that you grew frus..., you know, you felt weren't being productive, what, what happened during that period of time. Why couldn't you get your message across, cause you're obviously an articulate, forceful speaker. Why couldn't they understand what you were saying?

JW: Because, I believe—and I don't think it's appropriate to discuss this meeting or that meeting or who said what when a year, 18 months, 24 months ago—I believe that the reason that Autodesk did not change its strategies prior to Information Letter 14 was that the entire environment—the entire change in the market that was presented in Information Letter 14 had never really been laid out before. I think you'll find other people in the industry that consider it to be an expressive...

GZ: *(Interrupting).* But you could have laid it out privately to these folks.

JW: I could have, but then they would have to have taken it to everybody else in the company anyway, because this is a company that is not run from the top. It is a company in which innovation comes from the bottom, and if individuals in all of our business units are not aware that the expectations of the users

and the customers, of the opportunity in our market, if they're not aware of that, then they're not going to understand orders that are given to them from the executive office. This is a creativity business, and in a creativity business the people that are doing the creating have to understand the environment they're in. We've been a very, very successful company for a long time, and when you've been very successful with one formula, it's very difficult to change it. It's very difficult to understand that the world is changing. If you look at the companies in 1982 that did not make the transition from the 8 bit machines to the IBM PC, you see companies that were making so much money, doing so well with their old strategy, and the message was undoubtedly—and I think if you look at the history of those companies in many cases the executive suite knew what they had to do, but they just couldn't get the people who were doing the work to change their mindset. That's what we had to do. And in terms of what memo went to what person at what time, the important things to focus on is that Autodesk has, since April of last year, changed its strategy, has implemented virtually every... not in terms of implementing recommendations... but is addressing these things and is moving into these future with a strategy that is consistent with what will survive in this new environment.

GZ: I want to cover that, because they're a lot, they're a lot of things that, that, in specific I want to cover on that. The, the, uhm, now, when, when you, you're, you're, uh, some way through this ninety day period in which you'll be, you'll be back here, is that? Is that? What, what day are we on now, do you know?

JW: I don't actually count the days. I arrived in mid-January.

GZ: In mid January—so presumably you'll stay 'till mid-March, or is it a little open-ended?

JW: Uh, that would be mid-April.

GZ: I'm sorry, uh, uh, mid-April—would it, would you be staying—is it open ended, or is that sort of firm?

JW: Ahhh, it's firm to the extent that I expect to be able to take care of everything I came over here to do during that period of time. I'm quite confident of that and I think we're on track.

GZ: What... um, there are a number of companies in the software industry, some of them which you admire, namely Borland and, uh, Microsoft, which have, as their chief executives, uh, people who might, you might say, are, they are cut from the same the same cloth as your, as yourself. Technical people who have, um, a keen grasp of, of management and business issues. Um, it's served both Microsoft and, and, and Borland for the chief executives of those companies to to be, uh, you know, founders, who, who still, roll up their sleeves with, with programmers at their, at their companies. Why not you? What, what, what's... you would seem to be the ideal candidate.

JW: I disagree with your statement. You said, "You would seem... you seem to be cut from the same cloth." Every human being is different from every other human being. I don't believe there's much similarity between Bill Gates and Phillippe Kahn except that they both run highly successful software companies. I don't think there's much similarity between either of those two and myself. I think it's very convenient to put people in a box, saying this person is standard entrepreneur issue 1B—runs software company—I'm an engineer; I'm a programmer; I'm a technologist. I have no interest in running a large U.S. public company and I never have. It was the means to the end to be able to accomplish the technological work that I wish to achieve.

GZ: The, the sad fact is, is that if you look at the successful software companies and you look at the unsuccessful ones, uh, the Ashton-Tates and the Lotuses, they don't have that person there. The Borlands and the Microsofts, the Symantecs—they do.

JW: I disagree with you. The largest software company in the world is IBM. It isn't run by a software guy.

GZ: Would you be happy with their record? *(Laughs condescendingly.)* Would 'ya, would 'ya...

JW: Fifteen per cent compounded growth for a century? Yes, sir, I'd be happy with that record.

GZ: All right, their record in software, their record in software.

JW: Excuse me, excuse me. I did not say that, I spoke of their growth record...

GZ: *(Interrupting).* All I'm, all I'm, all I'm saying is that, uh, you know, you know...

JW: *(Interrupting).* Would you like to make a speech, or would you like me to answer your question?

GZ: *(Condescendingly.)* You know what I was talking about, I was talking about PC software companies, I was not talking about mainframe computer companies. Within the class of PC software companies of which AutoCAD clearly is a member, there are certain features...

JW: *(Interrupting).* It is not. It is not. It is not. Sir, it is not a member of the PC software companies. This is a very different business. You're likening us to companies that sell mass market products, at retail in a competitive environment, through mail order. We are an engineering software company, whose product is bought by professionals who pay ten times as much money for our product as they pay for the products of the companies you've named. And the model of our business... it is simply wrong to liken our business to Microsoft or Borland. We are like MacNeil Schwendler Corporation, like Swanson Analysis Systems, like PDA...

GZ: *(Interrupting).* What kind of CEOs do those companies have?

JW: They have experienced business managers who understand the operation of a company in a long-term, evolving market.

GZ: And is that your ideal for a CEO for Autodesk?

JW: I don't have an ideal for a CEO for Autodesk. Autodesk doesn't... and again, I think you are again making a statement that I believe to be unfounded in fact. I don't believe companies are run by Superman CEO's. I think that companies that are run by Superman CEO's live only as long as the Superman happens to do the job well and die with his first mistake. I think that successful companies are run by teams of people which include individual strong players who understand every aspect of the business. The CEO is the head of that team. But whether that CEO has a technical background, and has a marketing and a sales person and a finance person and a legal person on that team, or vice versa, does not matter.

GZ: *(Interrupting).* Is the team in place here at Autodesk, or, or do you think it needs to be augmented, and what way does it need to be?

JW: I believe we have an excellent team in place at Autodesk, and I think our record is indicative of that.

GZ: So, uh, I mean my understanding is, that, is, is, will Volker be staying on as the CEO, or is he an interim person, and if so...

JW: No, Volker is on an interim assignment as Chief Operating Officer, not Chief Executive Officer.

GZ: Okay, so, what, I mean, what, in terms of Alvar's replacement, then, what, what is your thinking about,

about that? You need someone who will just fit in, with the strong team that 'ya have already?

JW: *(Interrupting).* I don't need. I'm not on the CEO search committee; that's a board matter. And the board, is the board, which is charged by the investors...

GZ: *(Interrupting).* You are on the board, no, right?

JW: No, I'm not on the board.

GZ: You're not on the board anymore.

JW: I haven't been since 1988.

GZ: All right. *(Long pause.)* And what I'd say is, would, would, could you, can you imagine professional managers, of a stature that you'd want to be CEO, that would join this company knowing that you have a looming presence and a huge sway over the course of the company. Could they see it in themselves to, to see that as an asset, and as opposed to something to basically be afraid of?

JW: If they believe that, then I don't believe they should or would take the job, because if they believed that they'd believe something that isn't true. I don't have a looming presence over this company. I have an influence perhaps as a founder of the company as somebody who has been involved in product development, both in AutoCAD and in new products, but I believe you'll find if you talk to people, that between the time that I resigned as chairman in 1988 and the time I wrote Information Letter 14, I was not looming around behind the company, I was sitting at home developing new products, and that is what I intend to go back to doing in Switzerland just as soon as I can.

GZ: *(Long pause.)* OK. Uh, was there anything in particular that, that, uh, sparked, moved you to, to write this Information Letter 14, in the, in the, uh, in the letter itself you talk about how you don' wanna be seen as a stalking horse, but that there were folks within the company who you respected and who had made big contributions to the company who had, who had conveyed their concerns to you. Was that part of the reason, that prompted you or, or were the things you noticed yourself, or...

JW: No, I acted in what I believed to be the best interests of the company I own more than 850,000 shares of stock in. I wrote Information Letter 14 because I felt it was necessary, and the most effective way to protect, and to cause my investment to grow further.

GZ: I just meant that, in terms of wh, wh, why you did it then—was there any pre-, precipitating factors, or was it just a whole bunch of things, too, too, too many to, to number for me or to lay out for me?

JW: No, there wasn't any precipitating event other than I finished my previous project and decided to spend some time...

GZ: *(Interrupting).* You had some free time, to, to, look at the questions...

JW: Well, if you'd like me to answer your question, I'll be glad to; if you'd like to make a speech, go ahead. No, I had been working on the '286 version of AutoCAD and had turned that over to the AutoCAD development group at the end of, uh, 1990, and then decided to take some time to go out and look at the industry, buy some Windows software, learn a little bit more about the contemporary channels, look at the competition, and survey the competitive environment. Information Letter 14 was the result of that process.

GZ: Okay, okay. Had you used Windows before then, or, or had you been familiar with it?

JW: I had used Windows back in 1985.

GZ: Why, why does it come, I mean, I mean I'd have to say that I, with, with respect certainly to, to Windows, uh, uh, ya' know, I'd say I'd agree with you wholeheartedly, it seems like the company was late, late in recognising the importance of Windows and has been late to market with it, uh, uh...

JW: No, I disagree with that. We had a Windows project underway for both our AutoSketch and AutoCAD before Information Letter 14 was written. It's a major, job, particularly, well, in fact, we were one of the first to market on OS/2 with any product... and our Windows product essentially inherited all the work we'd done on the OS/2 product. It's not an easy job to put a product the size of AutoCAD under the existing 16 bit Windows. It's a major commitment of people for us, but I don't consider us late to market.

GZ: In the letter, you exhorted the company to move more quickly in the Windows area...

JW: That was one of many recommendations, but... I, I was actually quite happy with our progress with the Windows AutoCAD except perhaps in terms of the relative priorities versus some other platforms. I think...

GZ: *(Interrupting).* But that's the whole game. I mean, the relative priorities, is, is a lot of the game. I mean, you know, that's, that seemed to be what you were complaining about was prioritising and...

JW: *(Interrupting).* I think you must have read a different letter than the one I wrote. No, I was focusing on the marketing opportunities that we had. I felt that this was a product which, when done, should not simply be made available but rather aggressively marketed because we had a substantial opportunity to broaden the market. This is also not a one product company. We sold more than half a million copies of AutoCAD, in fact more than 600,000, we've also sold half a million copies of low-cost CAD, both AutoSketch and Generic CADD, and those are products that are in the retail channel—those are products which it was important for us to get on Windows as soon as possible and to increase our focus on Windows in the retail world.

GZ: *(Interrupting).* And so you were happy with the way that worked out, in terms of...

JW: I'm very happy with our position there.

GZ: Okay, okay, good. Uh, this, this, this one, one point, you thought I'd asked for, you, you, you said, "I've decided to take all the heat personally for putting these issues on the agenda. I'm not a stalking horse for anybody, and I don't want to suffer—I don't want anybody to suffer just because I happen to agree with them." Did, did you end up taking any heat for putting this stuff on the agenda, or...

JW: Not really.

GZ: What was the reaction, of, of folks to the, to the letter?

JW: Ahhhh, I think it was, uh, a little like when you fire a shotgun in an aviary, there was silence at first...

GZ: *(Interrupting).* In a, in a, in an "ave–ary"?

JW: In an aviary.

GZ: Aver-arary, okay.

JW: You know, where they have birds?

GZ: Um-hummm.

JW: There was kind of a period of silence, then a few little twitters began, and then a large amount of conversation. It had really the effect I intended it to have—it caused everybody to say "What is our strategy? What are we doing? What are the opportunities out there? What are the foregone opportunities that simply staying on our existing strategy may create?" And that's really the point. I mean, how can I possibly impugn the performance of the management of Autodesk when Autodesk has just finished the most successful year in its entire history? We have highest sales, highest profit, largest number of new products, greatest market share, broadest international distribution, more employees, more technology in the lab, more products scheduled for introduction next year than ever, okay. That's not a record of failure. The fact that our earnings were lower in the fourth quarter than some people on Wall Street expected because we sell to the two most cyclical industries in the world, construction and building,[399] and we're in the middle of a depression, is not anything that has to do with Autodesk strategy. We began this year with about a 70% market share in the CAD business, we ended the year with a 71% market share in the CAD business—71.4 I believe it is. That's not a company that's being clobbered by competition. That's not a Lotus being clobbered by itself. That's a company that needs to build on the base that we have, and expand that base with additional products, with additional platforms such as Windows, to broaden to the point that we are a broadly diversified company. That has been our goal since the foundation of the company. We have been in the multimedia market for several years; we're in the retail market—we're increasing all that focus as time goes on.

GZ: Is there, is is there any, I mean I sense from reading, uh, the, the letter that you feel the company should be in the class of a Microsoft or a Borland over the long run that, that, I mean, you talk about, you know, uh, the, the industrial empires that are kind of being created by these software companies and, and, and, I mean in terms of, in terms of a, a record compared to most industries, Autodesk's record is, is, is, is stunning but the growth rate has lagged at out some of the, of some of the, of, of, say the Novell, Borland, and Microsoft would be kind of the, you know, the triumvirant *(sic.)* of, of, of, of, of, real, real, real big ones.

JW: Novell's kind of a hardware company, still.

GZ: *(Condescendingly.)* Well, no, you can tell them that. Explain that to Ray. He'll be interested to hear what you have to say on that but, uh, the, uh...

JW: Look at their revenue. Their revenue has...

GZ: *(Interrupting).* All right, listen, so, so all I'm asking is that the, is there, do 'ya feel that Autodesk oughta be in that class so that your standard of measuring them, of measuring Autodesk, is, is, is that much higher, it's not just are you outgrowing uh, uh, the, the U.S. industry, it's are we keeping up with Borland and Microsoft and, uh, these...

JW: *(Interrupting).* I don't think we're keeping up with Borland and Microsoft. I believe that Autodesk.... I own no equities other than Autodesk, basically. The reason for that is that I do not believe there is any other company I can invest in which is better positioned to be one of the largest industrial companies in the twenty-first century...

[399] I meant "construction and manufacturing."

GZ: *(Interrupting).* My question was about whether, whether or not you feel any sense of frustration that you haven't kept up in growth rate with these companies, who, whose growth rate has been, you know, uh, uh, uh, higher than yours...

JW: You use "you" both to mean me and the company, which do you mean?

GZ: *(Interrupting).* You, you.

JW: Me, I don't have a growth rate, I'm an investor...

GZ: *(Interrupting).* No, I'm saying, are you frustrated that Autodesk's growth rate has not been as high as, as, as these companies I've mentioned?

JW: *(Interrupting).* No,

GZ: *(Continuing.)* And is that, do you feel, that, that, you know, it ought to be in that class?

JW: I believe that Autodesk, I, I don't use the word "ought"—I believe it will be in that class, and it will be in that class when we expand... when two things happen, when we expand the base we've built, and when the investments we've already made come to market. The fact that, for example, this year about 90% of our revenue is AutoCAD. Our...

GZ: *(Interrupting).* Is that up, down over the, over the, is that about the same as it's always been...

CA: About the same.

JW: I mean, over the long term it's down because it used to be 100%, but you're...

GZ: *(Interrupting).* Yeah, but over the last few years, over the last years, about the same.

JW: We expect that to be substantially less next year, perhaps at least to the say 87%—85% level. When you invest in something like multimedia, for example... we've invested in multimedia, we have I think a very large, perhaps commanding market share in the professional video production market. That's not a very large market, but that's a base that you can grow from—whoever owns that tools market is in the best position when the consumer explosion occurs in multimedia. We are not interested, and I don't believe we should be interested, speaking as an investor, in going head to head and battling it out in the C compiler wars, the spreadsheet wars, the word processing wars.... For everybody that enters that battles, only one tends to win, and there's a lot of losers along the way. Our strategy is very different. Find a market that is not mature yet, that is not developed yet, in which a relatively small investment can gain us a large market share, and a market which as the speed of computers, the graphics performance, the storage capacity inevitably increases, will become more and more viable and expand into a consumer market. That is precisely what we did with AutoCAD. AutoCAD... if you'd been interviewing me, as nobody did in '83, talking about "the failure of Autodesk in 1983"—we were just bumping along with a very slow growth rate and would have continued to bump along forever until the PC/AT came out. The PC/AT, the '286 machines, made Autodesk, because it gave us a platform with the power we needed. In multimedia it takes something like the introduction of the MPC. In scientific modeling, which we're entering in a month with the first desktop molecular modeler, it will take the 50 megahertz '486 or '586 machine to really make that. But if you get there first, if you own the market share in the market—if you own the market share in the market when nobody takes it seriously because it's so small, then you have the beachhead that you need to expand beyond that without making vast $100 million bets and taking away market share from Borland and Microsoft. That's our strategy. We control today... every

thing in this room, other than people, was manufactured. Every single thing was manufactured. The central part of manufacturing in the world today, today, is drafting, and we have an overwhelming market share in that process. That gives us the base at the centre of the design process. We can move earlier the design process into conceptual design, product presentation, rendering—on the tail end of the design process, into testing, manufacturing from that base. But we are in the drafting shop and, today, that is the point that every design passes through. If you look at those other markets in manufacturing, in architecture, in geographical information systems, they're easily distinct. That's a niche market. Yeah, that one's a quarter billion dollar niche market—that one's a half billion dollar niche market. Control of the the centre of the process, control of the design database, is what Autodesk—is what AutoCAD—gives us today. We can drop product after product after product into that market; we already have—our solid modeler,[400] which has sold more than 60,000—last week we shipped a major update to it—we have contractual arrangements with ESRI, who is the Number One player in geographical information systems, we're cooperating with SDRC, one of the leading players in mechanical design, and we are, through these arrangements, putting products through our channel, into our customer locations, that provide the solution that the designers and engineers need.

GZ: *(Interrupting).* Lemme, lemme askya, in that, in that regard because, because again from, from in reading about Autodesk in some of the, some of the, um, ex... uh, initiatives you have in different areas are quite intriguing, but, I'm, I'm kinda struck by a tension between being a, I mean, you're still largely a one product company, in, in terms of the revenue stream you have currently, and, and, I see, you know, there's, and, and a lot of the things you're talking about are basically leveraging off of your core strength in *(grunts)* in design software and how do you, how would you, uh, describe to me wh..., where, uh, uh, is, is it, are you the design software company, or are you a, you know, software company that just happened to have, hit this mother lode in this one design area but you really are a general, you know, a general diversified company that you plan on having, you know, ho– hopefully, you plan on on hitting something big in some rela...—area that may not be related to design, or is design pretty much the thing that, that pretty much knits it together for 'ya? And anybody can jump in and answer that one all right, I mean, but, but, uh...

JW: Well, I don't know which of the eight questions you've asked you'd like me to answer first, but, I think when you characterise us as a one product company, we're an unusual one product company, in that most of the other one product companies you've named have not had the retail product of their product rise from $1000 to more than $4000 over the last ten years. That rise has been the consequence of additional capabilities being put in that one product which...

GZ: *(Interrupting).* I'd like to talk about pricing, separately, but, but this question, this question of whe...of whe...of, of where you're heading, is it, is it, to be a design software company or, or to be, you know, in...

JW: We intend to be the leader in design software, and we intend to be a significant player in other industries including multimedia, including scientific modeling, including information systems. And if you look at our business unit organisation, each of those business units is chartered as being one of the top players in that area,

GZ: Okay, let's, let's talk about that, that, that, uh, that, uh, that, not in great detail, but, but, briefly, in the information systems area, all right, you're referring to the, on line, uh...

JW: Xanadu and AMIX.

[400]The Advanced Modeling Extension (AME).

GZ: AMIX, right, right, and uh, the multimedia, you've got products in that, and you've got more coming.

JW: We have three products already launched and more coming.

GZ: Okay, and the scientific modeling, you've got this product next month? Is that the int... is that the first product in that, in that...

JW: And in our retail products division...

GZ: Yeah, *Chaos*, already, right, yeah...

JW: Course, that's over on the edge because we've had a retail products division in Bothell Washington that sells Generic CADD, the Home Series, AutoSketch, and also distributes the scientific products Chaos and CA Lab, and they'll be doing additional retail products in various areas, not necessarily limited to CAD in the future.

GZ: Let, let, let, let me ask you about, uh, information and multimedia, uh, uh, again just, just briefly because, I mean, these are big subjects, we could spend all afternoon just on them, but, I, I'd didn't want to throw things, throw out some, some reactions I had, just, just some ideas, uh, I mean, the parent company for, of *The Wall Street Journal*, is in the electronic information business, Dow Jones News Retrieval, I don't know—I don't know if—presumably you've used it, uh, uh, ah, what can you bring to, uh, bring, to, to that kind of business that the current players who, you know, are really, look, you know, are really, look, you know, leveraging all that information that they've got—proprietary information or, or maybe they've brokered it, in case of Dow Jones, you know, they have a whole, they're, they're, they're, they broker information for, that, other people, that other newspapers and publications have... what, what's, is there something unique you can bring, or, what, what's, what's the the formula there you're, you're, you're, you're working with?

JW: The most powerful thing in the world—a new idea. What, if you're talking about AMIX, which is the online service component of what we're doing, AMIX is to Dow Jones News Retrieval and CompuServe as financial futures are to a bank. In your own newspaper, nobody took financial futures seriously until they'd established a strong beachhead. You could say, well, there's forward contracts and banks and they've been around for fifty years, and yet suddenly not only did an entire multi-billion dollar market appear out of nowhere, it now uses up a large part of the back end of your paper. What we are doing is creating the first information market, and that is something that is very fundamentally different from selling the kind of online services...

GZ: *(Interrupting).* Yeah, I don't actually, and, and I mean, because, because I, I can't grasp it, but I don't see that distinction. I mean seems to me that, all, uh, you're distributing information... you can... marketplace seems like a nice marketing idea, but it's fundamentally the same kind of business as, as Dow Jones News Retrieval would be.

JW: No sir. I would recommend you read their brochure and the background...

GZ: Give it to me in a nutshell. What, what am I missing? ...that I don't, that I haven't grasped it... what am I missing?

JW: Your statement is, is wrong.

GZ: *(Interrupting).* Why?

JW: We're not distributing information in any fashion. We're providing a place where information buyers and information sellers can do transactions and we make our money on the transactions. We are not setting prices, we not deriving contracts, we are providing a market...

GZ: *(Interrupting).* So you're something like, like an, like an auctioneer of information, or, or...

JW: *(Interrupting).* We're like nothing you've ever seen and if you look, at for example today, at the market for market research, in PC trends, in financial research, the entire market for financial market newsletters, the market for consulting, in any one of these fields there's an extremely inefficient market. It costs the consultant—he spends half his life marketing himself. If we can allow himself to market himself and market his product at a cost which is a small fraction, we believe we will create an efficient market in expertise.

GZ: So, it's geared towards, then, not publications then or published information, but sort of the information locked up in peoples' minds that maybe consultants or advisers, is that...

JW: Published information in the sense of products—we can distribute software for $5 and have both the seller and buyer and us make money on it. That channel doesn't exist today...

GZ: Okay, electronically...

JW: Electronically...

GZ: Okay, all right, so but, but I should think of it more in terms of the expertise of these consultants or software, rather than, rather than, another, another vehicle to, to get people to buy a magazine, essentially—the contents of a magazine.

JW: That is one unique aspect of it. On Dow Jones News Retrieval, for example, you cannot, as you can today with AMIX, if you open an account, request consulting from Mitch Kapor—he was our second customer. Mitch is on the system; he will do consulting for you. You may not be able to pay his price. As additional people with expertise come on that system, we will be providing a vehicle for anybody who can pay the price to call on industry leading figures...

GZ: *(Interrupting).* And so it create efficiencies for them in terms of finding customers.

JW: They don't have to spend all their time making phone calls marketing their expertise.

GZ: Let's talk about multimedia a little bit although, uh, uh, how important do 'ya think uh, content is to the whole uh, equation of this and, again, anybody can jump in on this one, I mean, I mean, the, the, you've got Bill Gates thinks content's important enough so that he's trying to buy the rights to electronic, uh, art, I mean, uh, from my talks with Apple, uh, they seem to think content's im- im- im- important in the sense that, uh, say, uh, I don't know if you're familiar with a company called ETAK, uh, geographic, geo-coded information, you might just want to buy it all, the information, the geo-coding and just slap into whatever device you've got there. Do, do you, um, or you could be in, in the software tools business, you could be the kind of Adobe of multimedia and, and make something that just isn't needed in, in, in the process, but um, whu, whu, whu, I happen to think myself and this, this again may be, may be premature to come to this conclusion, but the personal productivity business for this multimedia is, I think, going to be the smallest part of it, and, and just wondering, are, do you see the content part of it as, as, important.

JW: How do you define the "personal productivity part."

GZ: Well, forget I said, that, all right. Let's just, d'ya, d'ya, d'ya, d'ya, d'ya, see content as important to the equation in multimedia, or, or is it a, is it—can you pursue it in pure software, as a pure software opportunity.

JW: Well, content is software.

GZ: *(Condescending, like your second grade home-room teacher.)* Well, what I mean by software is com-puter pro-grams, and what I mean by con-tent is stuff that most people think of as artifacts,[401] films, books, pictures, you get the idea, okay, so, so that's what I mean—I don't mean software in the, in the sense that Sony talks about it, okay?

JW: Well, I believe that there is, as multimedia plays out, there will be plenty of money to be made in every sector of the business. I think if you look at Sony's business, for example...

GZ: *(Interrupting).* But I asked you, if you needed, if you needed content, okay? Do you need content?

JW: For multimedia to become a large, retail, consumer business, content is necessary.

GZ: How do you go about getting it?

JW: That's not necessarily the sector we'll choose to address...

GZ: Okay, that's what I'm asking; so you're not going to address the content...

JW: *(Interrupting).* I did not say that. You're putting words in my mouth again. What you asked me, if I could answer your question, is "do you believe content is important?" Content is very important, but the people that make VCRs do not necessarily also make the tapes.

GZ: *(Interrupting).* Okay, so let's move from that, since you, since you've agreed that content is important do you think it's important for Autodesk to be in the content business?

JW: Perhaps. Our director of that business unit isn't here, I shouldn't put words in his mouth, but I believe Autodesk can make plenty of money purely selling tools. I believe we can probably make more money selling content, but many of these so-called content businesses tend to be very low margin publishing type businesses rather than high margin businesses and you have to analyse it from a business standpoint. If the dynamics of your business end up looking more like a record company than a software business, you have to say "how does that fit our business model?", "how does that fit our distribution channel?" It's not clear to me that the content—the content may be sold through record stores rather than through the software channels that we have. And one has to analyse those questions; it gets extraordinarily difficult when there isn't any market at all out there now. It's like the early days of microprocessors; Intel was selling primarily development systems rather than chips. You have to seed the market with the development systems; you have to get the tools into the hands of the content creators before the market begins to develop.

GZ: Do you have any kind of time frame within your own mind, any kind of gut feel for, how, that, how long that development process will take. Is it five years, ten years, two, or...

JW: Well, of course, we're not really in control of it...

GZ: *(Interrupting).* I know, I just asked for your gut, I just asked for your gut sense of it, though *(conde-*

[401] ?

scendingly) I know you're not in control of it.

JW: I believe that within five years we will see substantial consumer presence at home. But, understand, I don't believe there's going to be this thing called "the multimedia market" any more than there's a thing called the "graphics market" today. There will simply be a time when every computer has sound capability and video capability and it's integrated with everything else the computer does. You aren't going to buy a separate multimedia box; it will simply be part of what you use in a business environment or a home environment. These pieces will come together.

GZ: Okay, okay. Uhhhm, you, you made an interesting reference in, in this to the, to the Cyberspace project which, I, I think you refer to in one of the story because I did mention it, uh, uh, the Autodesk project in a story I did on, on virtual reality, but, how, how, uh, you know, uh, I looked at... the, the point of, of getting attention for the company doing something cutting edge like that, uh, I think is well, well taken. What, whu, whu, whu, what, what do you, uh, uh, I've been trying to talk with people and see where the commercial prospects really lie for virtual reality and, and uh, you have one of the project that seems farthest along in terms of commercialisation.

JW: We'll be shipping in third quarter.[402]

GZ: Yeah. Do, do, do you, I mean, do you see, is this the, you know, as some would have us to believe, the, uh, uh, you know, a whole new way of relating to electronics, or is it more something that's just going to be incremental, sorta like multimedia, just end up being kind of an aspect of the computing experience but not necessarily, we're not going to necessarily, uh, uh, it's not necessarily gonna be a, an industry its, its, in itself, uh...

JW: I believe the latter; it will be an aspect of the industry. In the Sixties there was what was called "the computer graphics industry" and it was companies like Adage and so forth who sold graphics devices. And today every machine Sun sells has graphics in it, but it's not a graphics machine, it's just that graphics is the way you get at the machine. In virtual reality, eventually, I believe, elements of that technology will become mainstream parts of the way people interact with computers...

GZ: *(Interrupting).* The glove, for, um, is there anything you can single out? Is the glove one of those features you think is...

JW: Well, I'd rather not talk about specific pieces of hardware since it's kind of talking about mice versus trackballs in 1968—I mean, the technology is a way to provide more direct human interaction with computers, in the same sense that the mouse was an improvement over the keyboard. Whether the glove, whether infrared head position sensors, whether goggles you wear, or LCD screens, whatever, the technology is not the point. We're struggling along with extraordinarily crude technology today; we're about where Xerox PARC was with the Alto in the mid-1970's. They had something that was extraordinarily expensive, made only in tiny quantities, that they were using to work on how you would use—how people would use this technology when it became widely available. Not until the Macintosh, almost 10 years later, was that technology available at an affordable price. I don't think it'll take ten years for at least the first elements of virtual reality...

GZ: *(Interrupting).* What...the product that you guys have for later this year, is, is it predicated on any particular hardware?

[402] After the Advanced Technology department was disbanded in August 1992, Cyberspace was transferred to the Multimedia group, who rescheduled it. It shipped in February 1993.

JW: You can use it with a mouse and a VGA.

GZ: Oh, okay.

JW: You can enter with the PC hardware you have now, and it will support everything up to the highest end.

GZ: Okay, is it, and, and, and, and, and what, what, what, what, a lot, uh, makes you define it as, as, cybersp, cyber, virtual reality? Is, is it, is it, I mean, if we're not using any special headgear or special, uh, uh hand hardware, what, what, what...

JW: *(Interrupting).* Well, that's not virtual reality. Virtual reality is the emulation of a three-dimensional environment inside the computer.

GZ: Okay, so, and so how are you doing that—how are you achieving it if, if not, if not through the means of, of uh, of the hardware?

JW: The hardware has nothing to do with it. Virtual reality—a 3D CAD system is a primitive virtual reality system.

GZ: *(Interrupting).* Yeah, I guess. *(Louder.)* I guess! I mean, I guess!

JW: I believe that...

GZ: Okay.

JW: What our Cyberspace does is allow the definition of a virtual world and then allows interaction between human being or beings and the objects within that world. You can put on the headgear, you can walk around inside the world. You can climb and you can you can pick things up and drop things, or, you can do the very same thing with a mouse and a VGA. It's a lot more real if you do it with the headgear.

GZ: Well, if you do it with the mouse and the VGA, how is it different than the CAD software you have out there now, uh, what, what, I mean, whu, how is it different?

JW: Because it knows about physics amongst other things—the fact that objects collide with one another, that there's gravity in the world, that things deform when you pull on them. It's a model of the world.

GZ: Okay, okay.

JW: And I suppose you could say that the very highest end CAD systems have that capability also, but that's the very, very high end of CAD and none of them really provide access to that—to the world builder. They're really concerned with the, engineer...

GZ: *(Interrupting).* Do any of your systems qualify in that very high end that you're talking about, then?

JW: AME, yes...

GZ: AME, would, would, would have some of that knowledge of the world...

JW: Yeah. It knows about differences between objects and intersections and clearance fits...

LG: 'Ya ever seen this, um, this product?

GZ: *(Pause.)* Uh.

LG: You ever seen *(unintelligible)* uh, okay...

GZ: Yeah, yeah, yeah, I, I haven't seen a demo of, of the Cyberspace product but I have seen a lot of VPL's stuff and NASA's stuff and the stuff at UNC, ah, but, uh...

JW: You've seen AutoCAD?

GZ: Yeah.

JW: You've seen AME?

GZ: No, no, that I haven't.

JW: You've seen our multimedia products?

GZ: Uhhhh, maybe I have, maybe I haven't, doesn't stick in my mind.

JW: Generic CADD?

JW: *(Pause.)* AutoSketch?

GZ: AutoSketch, well, I can come back for a demo of that one. I mean I'd certainly be, I'd certainly be, you know, glad to do that. I mean, this, this, this was kind of uh, uh, so why don't we do that.

JW: Home Series? I mean, we've shipped thirty thousand of them.

GZ: *Mea, mea culpa*, okay, *mea culpa*, I'm a Macintosh.[403] Actually I'm a, a big Macintosh user and so, I, and I don't, I don't use your Macintosh CAD so, I would be most familiar with Macintosh software, but, uh.[404] Okay, all right, so, uhhhhh, all right, well what, eh, it, it, it, it, can, can we, uhhhh, uh, break down, I guess to the, to the essentials of, ah, where, what are some of the changes that have been made over the past year, why you think they'll, uh, you know, help the company and maybe, maybe we can go through some of them and just discuss a little bit why they're going to benefit the company.

JW: Would you prefer to direct that to me or the people that are making the changes?

GZ: S, sa, same, you know, same thing. Same thing, I mean, just jump in, I mean, just jump in, you know.

AG: We're looking at, you know, certainly in the early part of the year the reorganisation was probably the news at that point, and what we've done, if you look at the history of the company, we've grown from a small company to still a centralised large company. And what happens in an environment like that, you generally get displaced authority and responsibility. I mean, the people can make decisions in the company, but they may not have the corresponding responsibility for those. This was an attempt to put the authority and responsibility of each product in that product group, so rather than a person at the top essentially controlling what happens with each product, you really put the authority and responsibility in the hands of the people doing the job. What that has done... it has essentially pushed, once again, the authority and responsibility down the organisational chart into the hands of the people *(inaudible)*.

GZ: Is, is there any tangible, so far, you see from that?

---

[403] I find this hard to believe. In my experience, Macintoshes are friendly, straightforward, and unthreatening.

[404] Here is a guy who had been beating Microsoft and IBM with a baseball bat for two years, now admitting he was a Macintosh user and didn't know very much about PC software.

AG: Uh, you know, I, I, certainly, you know, I'm sure Ruth can can attest...

RC: *(Laughs.)*

AG: But, you know, once you, uh, what happens with a typical organisation, with, with marketing, finance, product development, you always, no matter what the company is, you end up with communication problems between the departments. What this does, it puts all of those functions into the hands, once again, of the people who are doing the job. The people who produce the products are now responsible for marketing those products, they're responsible for coming up with budgets for those products, they're responsible for the profitability of those products. And that's what we're seeing: we're seeing the responsibility pushed down into the people...

GZ: *(Interrupting).* All, all I asked was, do you see any tangible benefits from that, is there anything you can point to?

JW: A tangible benefit is that every single business unit is delivering a product this year, with a launch plan that was developed within that business unit. We're not shortchanging the smaller business units in favour of the AutoCAD business unit. If that business unit is bringing in 3% of the revenue, it should have 3% of the resources. Not fifty, not zero, but something appropriate to the size of its market and our commitment in it. And we've been able to delegate to the heads of those business units the authority, the power, the money, the people to get the job done, and they're getting it done.

GZ: Is there one example you might single out of a particular business unit that you're either especially pleased of, the way it's responded to this kind of...

AG: *(Interrupting).* Actually, I think all of them, every single one.

GZ: *(Interrupting).* Name one. You, you name one *(unintelligible)*. Well, see, you seem, you're on, you're saying, I mean, I obviously, I cannot even contemplate trying to mention all of them so, so, what, what, what comes to mind, is, is, is one worth mentioning.

RC: Well, obviously, as the only business unit represented here, AutoCAD, one of the things that the reorganisation has done for the business unit is really allowed us to bring the product marketing and the technology people together as a group, to really...

GZ: *(Interrupting).* They used to be separate.

RC: Yes, like many other organisations that...

GZ: *(Interrupting).* They were functionally organised. And so now, and so now you have, R&D people within your, your business unit, or...

RC: We have R&D, we have technology, we have marketing, and we many of the other traditional functions which you find in a company.

GZ: And so you're, sorta of like a, uh, uh, uh, a CEO of a, of a company, in uh, in effect, right, cause you obviously, well, except for the finance arm?

RC: We have our own finance, also, within the groups, so that allows us to really take a look at what is happening in the marketplace, really review any trends that we might see happening in the marketplace, take a look at what's happening with sales, and also to go out and move much quicker as far as what we

may have done and the implementation of some technologies that...

GZ: *(Interrupting).* When did that take effect? Uh, for you?

RC: That was the first of July, and we had been working on it for a couple of months prior to that.

GZ: Okay, okay, and, uh, uh, so you got about nine months, you got close to, close to eight months under your belt.

RC: A gestation period.

GZ: Uh, can you see, I mean, uh, are there, can you single out some, you know, tangible benefit about, I mean, are you seeing faster results in product development?

RC: Definitely. Definitely. We are getting much faster results. Time to market for product now has increased, I would say, by a good 50%.

??: Decreased.

RC: Sorry, sorry guys!

GZ: What was it taking you get a product to, to market?

RC: That might vary, but by having all the people working together in teams, or smaller teams, we've really been able to cut down, cut across a lot of functional lines. And the result of that really is the number of products that we're going to be shipping this year.

AG: More of a parallel development stream.

RC: Correct, correct.

JW: Yes, we had never worked on, for example, Release 13 of AutoCAD until we finished Release 12. And Release 13 is actually well along the road now because the AutoCAD Business Unit has the resources that it can spread among...

GZ: *(Interrupting).* That's what you mean parallel developments, okay, so you're working on not just the next version but the one after that at the same time. What, you, you, Al, come on, you're just about to release 13...

JW: Twelve.

GZ: Twelve. But, so, you presumably now have fourteen under development as well, is that...

RC: Thirteen.

GZ: But you don't have fourteen under development.

JW: Well, *(sputters).* Major components of fourteen are the underpinnings of thirteen. We're about to...

GZ: *(Interrupting). (Shouts.)* Well, maybe, you just gave me as an example, I'm not trying, this isn't a trick question, I'm just presuming since you said you now do two of 'em uh, that, I, I can't...

AG: We're eight months into this. Correct me if I'm wrong, but we're already six months into Release

thirteen, which means that six months ago we had twelve and thirteen going in parallel. And we're now six months into that: Release twelve will be coming out mid-point of next year, at which point we'll be a year into the development of thirteen. Under the old environment we would have been starting thirteen at that point.

GZ: Okay.

JW: And not even at that point. We would really be starting thirteen not just after the first customer ship of Release 12, but after...

GZ: *(Interrupting).* After the bug fixes...

JW: Not so much bug fixes, but remember AutoCAD runs on the Macintosh and the Sun and the DecStation...

GZ: *(Interrupting).* And these were all the cross-platform. so that really slowed you down, hah!, that really slowed you down, okay.

RC: And, and I also want to address John's point. Many of the things we're working on in Release 13 really are the underpinnings for Release 14, 15, and the CAD system of the future.

GZ: Okay. So this is a major upgrade in the, that sense, too.

RC: Absolutely.

GZ: Yeah, okay, well that's a good example.

CA: I have another example too. Retail Products, I think, is worth mentioning.

GZ: *(Interrupting).* What's, what's your name again, uh...

CA: Carolyn Aver.

GZ: Carolyn Aver, I'm sorry Carolyn, yeah, uh...

CA: Okay. Our Retail Product division comes from our Generic subsidiary. We bought Generic Software, pretty much left them alone, within the Generic product line. They did take on responsibility for Auto-Sketch along the way, but for the most part it was really focused on Generic. Having really given them the full responsibility of a business unit, they have made numerous presentations to the management group and we've really endorsed them to become a fully retail products, uh, business unit. They've come out—one of the things John alluded to before—is the Home Series, which is a low-end $59 product that focuses on baths, or kitchens. Those products have done extremely well. They'll be coming out this year with a number of products that are completely outside of the CAD industry...

GZ: *(Interrupting).* Oh will they?

CA: Yeah. We can't tell you what they are...

GZ: *(Interrupting).* You can say what, what areas they are, okay...

CA: They're working on three different products, come to mind now...

GZ: *(Interrupting).* In this, the, the, the, the Generic Software, you, that's the result of an acquisition, that you, that you made, that's run as a...

CA: *(Interrupting).* That turned into Autodesk Retail Products.

GZ: Turned into Autodesk Retail Products. But, and when was that made, that acquisition?

CA: The acquisition was made in '89, '88–'89,[405] and as part of the restructuring that turned Retail Products. So they've really changed the direction of that organisation over the last six months because they've been given the ability to run as a business unit. They're projecting revenue increases of over, I think, 75% and are really planning on building a pretty big market out of that, so there's another thing that...

GZ: *(Interrupting).* Okay. And moving into areas outside of, of, CAD. Okay.

AG: Going back to the original question of what has happened in the reorganisation—another large part of that that we're seeing results from right now.... Last year we went into the reorganisation of Europe. Before, we were operating in Europe out of a subsidiary in England, Switzerland, and Sweden. Now what we've done is we've set up offices in virtually every country of Europe—the reason here being that we now have a direct staff in those countries that are supporting application developers, that are supporting the customers, that are supporting the dealers. This was always removed—one level of distribution removed because that was happening from Switzerland. So once again we've put more staff, more capital into each of the countries in Europe. We're seeing quite a lot of results from that. Probably the most obvious is Germany. Germany, Italy, Spain are countries that are doing incredibly well as a...

GZ: *(Interrupting).* What's the, what's the overseas sales, uh, the, the, the breakdown of revenue, uh, is it...

CA: About 55% international, 45% United States.

AG: Now that's international—that's not just Europe.

GZ: Yeah, yeah, but so, so you're, you're basically hiring more nationals or more folks within these individual markets and...

AG: We always, even when we had subsidiaries in Japan, in Scandinavia, Switzerland, we always hired nationals but, yes, as we're going into each country, we're staffing them entirely with nationals of those countries.

GZ: Have you been adding more people in Europe, I mean, at a higher rate than you would in the U.S. or, or, uh...

AG: We were last year, and...

GZ: *(Interrupting).* You were last year, okay, and you've got to the point you wanted.

JW: And that saw a virtual double in sales in several of those markets last year.

GZ: Did'ya? Yeah, I mean, I mean, I mean, Germany in particular, I mean...

CA: Germany, Italy...

JW: Spain.

CA: Spain.

[405] March 7, 1989. See page 489.

GZ: Doubled. Last year.

CA: In local currency terms.

JW: Just about.

GZ: Well, with dollar.... Locally doubled.

CA: In local currency. *(Laughs.)* Sorry, still all the caveats of *(unintelligible)*.

AG: But still again, last year, last year, what I'd like to point out...

GZ: *(Interrupting).* Are you talking about the calendar year, or you're tellin' me 'bout, or the fiscal year...

CA: Fiscal. We're a January.

GZ: Talking about the fiscal year. Okay, okay. Your fiscal year ends January 31st, so, so it's roughly, it's roughly, it's roughly the year 91 we're talking about when you say it doubled. Okay.

AG: But last year, going through that restructuring, had an impact on our profitability, because we believed it was something we needed to do to expand that market and help revenues. That did, in fact, happen and we got the benefit of that this year. So, you know, that's part of the reorganisation that's...

JW: And we're now doing that in the Far East and we've reorganised the United States—the Americas—Malcolm's division, on that model. So, we now have a clear division of the company in the product related activities which the business unit managers like Ruth are responsible for and the geography related sales/customer related activities which are organised in Europe, Americas, and Far East. So we have a much more manageable structure in terms of what the CEO sees, in terms of who's responsible for each territory—who's responsible for each product. In each one of those heads, has essentially complete discretion to run their territory or their business area within budget.

GZ: Okay, okay. Uhhm, okay, uhh, let me hop around on a couple of things because, uh, I'm *(cough—unintelligible)*, the, the whole issue of, of AutoCAD's pricing uh, which was, uh, uh discussed some in your memo John, and I, and I, don't know where that, that stands but, but have you, have you, uh, taken steps to, ah, I, ah, to, to, uh, respond in some way to, to, to, your, your, your, your concern that maybe, maybe it's priced too, too high or the premium is a little too much of a good thing, or, or...

JW: I don't believe it's priced too high. I don't believe I said that. Again, I come back to my statement that there is a difference between a retail product and a professional product that's a lot more than just the price. If you look at the half million CAD systems we've sold at $600...

GZ: *(Interrupting).* What you did say was, "It would seem wise to revisit the question of AutoCAD's price and ask whether it is consonant with the pricing of software products which will maintain and expand their leadership in the 1990's." So, did you revisit the question and what, what did you come to the conclusion...

JW: We did, and we found that AutoCAD was encountering no price pressure whatsoever in the market. In fact, in Japan we sell AutoCAD for more than twice the U.S. price, in Germany we sell it for close...

GZ: *(Interrupting).* Hah, hah. Well, they deserve it, uh, but, uh...

JW: Excuse me, let me finish, because I'm about to say something that's rather important. In Germany we

sell it for close to twice the U.S. price, in India we sell it for a fraction of the U.S. price, in the Soviet Union we sell it for 25,000 roubles, whatever that's worth today—we price the product based on where it has to be in that market based on what engineers are paid in those markets, based on what companies can afford in order to carry out our market share strategy. In the United States we are not seeing competitive products take away sales from AutoCAD based on price, so in that environment there is no reason at this point to change the pricing of AutoCAD.

GZ: Okay, okay. Uh, the, the, another thing you mentioned in there that, that sort of relates to, to pricing and, and, uh, I wanted to ask 'ya about was, was the, the uh, the nightmare scenario, of Microsoft, uh coming up with, uh, I think a nicely named product, "Windows Engineer," uh, and, and, as you, you know, within a share, within a year their market share exceeded fifty percent, I, I think, that is a nightmare scenario, I, I don't think Microsoft could, could make anything go off that well, but, but do you have any sense that someone, maybe, that some other company, whether it's Microsoft or, or other big company could, I mean, actually has that intention to jump into this market?

JW: I know of one big company that does. That's us. And in March we're shipping AutoCAD for Windows at a $99 premium to all of our installed base and we intend to have a lot more than 50% market share by the end of this year.

GZ: All right, but is there any, you know, is, has Microsoft, I mean, has Bill, Bill actually admired your memo, as I know you, you must know and, and did he take your advice? Did Phillippe Kahn take your advice. Did Jim Manzi take your advice?

JW: As far as I can tell, in Bill's memo he said they weren't interested in markets like that. I think that the wise people in this industry such as Phillippe and Bill and Jim Manzi recognise that there are difference between retail mass market products and professional products and that one of the things that they would have to do is not simply put the product in Egghead, but rather build the kind of dealer/distribution/support channel that Autodesk has spent ten years building. That's the way professional products are sold and people who have taken the retail tack... we've had retail competitors in lots of markets including companies such as IBM over the last ten years, and they've never been able to crack the distribution channel. We're not just talking about CAD software—we're talking about the other things these professionals buy like drafting tables, like high-end plotters, are typically sold through a dealer environment. And we have, at this moment, complete control of that dealer environment.

GZ: So you don't have any sense, that there's anything brewing then...

JW: I do not.

GZ: Okay, okay.

MD: There are always things brewing. There are any number of competitors out there that would love to dethrone us. I don't think there's anything brewing in terms of a competitive threat that's...

GZ: *(Interrupting).* Uh, uh, from someone the size of a Microsoft, or a Borland, or uh...

AG: Actually, in the history of the company, the nightmare scenario is the one where we're competing against IBM, Computervision, McDonnell-Douglas, and everyone else in '84–'85, I think at that time, if you wanted to lose sleep, that's the...

GZ: *(Interrupting).* Yeah, those are the people to worry about, yeah.

AG: And they all came in with very competitive products. I remember Crossroads by McDonnell-Douglas looked as though it was aimed right at our heart but, once again, the whole issue revolved around the distribution of the product.

JW: And also, the customers invested in it. If you had put the entire asset base of your architectural practice, or the drawing base of your factory into AutoCAD, and you've trained all your people on AutoCAD, you have to have a compelling reason, and I mean, let's face it, this is a world in which there's such a thing as software piracy, and the fact that people will pay $3500 for another copy of AutoCAD versus just stealing one indicates that the channel is delivering productivity to them—that it's worth paying for.

GZ: Given that, were, were you engaging in some hyperbole then, uh, in this section of the memo, trying to, uh, maybe, I don't know, what, just spark concern, where maybe there wasn't, there wasn't, uh, or spark some sense of urgency, is that, what's the idea to be, sort of, exaggerate the straits the company was in?

JW: Ahhhh, no, I was not attempting to exaggerate the straits the company was in, I was attempting, in titling it "The Nightmare Scenario" to describe a situation the company could be in if the company did not have a strong, well-positioned Windows product in both the retail market and the professional market. We will be launching both the retail product and the professional product this year. The retail product—the first retail product will be Sketch for Windows, and that will be shipped in second quarter.

GZ: And that's the first Windows product the company will have shipped?

JW: It will actually be the second because AutoCAD will be the first.[406]

GZ: And when is AutoCAD coming...

RC: March 1st.

GZ: March first, okay, yeah my question at the beginning about this late, late for Windows, I mean, I mean clearly, ah, ah, of the, of the major category leaders, uh, as far as I can tell, you're the, you're the, you're the, uh last to come in with a Windows product and that, that, that, that's the only sense in which I meant it was late. You might not be late according into your own internal compass, but, but, uh, has it it cost 'ya much to be, to, to be coming out in March as opposed to, uh, six months ago or...

JW: Well, first of all, in terms of category leaders, it will be the first professional CAD systems to come out on Windows...

GZ: *(Interrupting).* Okay, but I meant, you're the leader in that. I'm comparing you with other, with other category leaders.

JW: In terms of cost today, Windows, 16 bit Windows, Windows 3.0—3.1 is not a competitive platform for CAD. It is substantially slower than a raw '386 machine, and not until...

GZ: *(Interrupting).* Just running, running the, running, the DOS version...

JW: Running the DOS version which runs in 32 bit mode.

GZ: Okay, okay.

[406]I misspoke here. Actually, Autodesk Animator Player for Windows was our first Windows product, and it had been shipping for over a year at the time.

JW: When Windows NT or 32 bit Windows comes out, then Windows will be a competitive platform. At this point, until that happens, professional CAD under Windows will be a relatively painful thing to do. Our goal is to get the product into the market. And there are people who can benefit from the inter-application operability, cut and paste...

GZ: *(Interrupting).* You, you don't see a, a wholesale shift to your Windows products...

JW: We expect a wholesale shift to the 16 bit product which will then turn into a flood when 32 bit Windows comes out. We will have a 32 bit Windows product at the same time 32 bit Windows becomes generally available.[407] That's the point that Windows will become a workstation platform comparable to Sun or the new Macintosh with System 7. Everybody has to wait for that, and getting there before that happens is really getting there early, in a sense.

GZ: Uh, so, lemme say something on this general subject of Windows cause I, cause I cover Microsoft and, probably pretty intensively and, and I've been around the block in '88 and '89, in particular, late '88—uh, late, late, late '89 and early '90, about, uh, Windows and its future and what it meant for peoples', uh, application strategies and, uh, I mean, uh, how would you guys characterise yourselves... were you, were you, wait-and-see, doubters, uh, were you ju, on the bit, Windows bandwagon but, like you said, for technical reasons it didn't pay to, uh, really push it, uh, what, what was...

JW: We were, really there, depending on how you look at it, we were there before everybody else, but in a different way. We were one of the very first on the OS/2 bandwagon...

GZ: *(Interrupts, snorting derisively.)*

JW: Let me finish—let me finish—let me finish. If you'd like me to.... Windows and OS/2 are, from an application programmer's standpoint, essentially the same system. You change some of the function calls, you change some of the guts, but in fact they are basically the same. So, at the time... Windows was not a viable platform for CAD, I would say not for anything until Windows 3.0. Okay, so we were up on OS/2 1.0 long before Windows 3.0 came along. So we had the API calls, we had the dialogue box interface, we'd squeezed all of that into the memory model, all of which come directly over to Windows.

GZ: Okay, so you were able to recoup the investment in that OS/2 work.

JW: We were, in fact, running under Windows 3.0 a couple of weeks after we got the System Development Kit. Going from that point to having something that performs, is professionally packaged, has the appropriate documentation and help file—that takes time. But we didn't lose any of our investment is OS/2, and it got us on Windows much sooner than we would have been there otherwise.

GZ: Okay, well when did 'ya, when did you, uh, I mean th, th, there are points, points in here where you reflect about, you know, can we, can we throw more, you know, get, be more intense about this, did, was there a point in time when you decided, yeah, we've just gotta ratchet up the energy level that we have behind the Windows AutoCAD project? When, when would that have been?

RC: As soon as we saw that it was going to be a commercially viable product, and that it was going to bring some...

GZ: *(Interrupting).* So sometime soon after May, of, of 90 then, uh...

[407] We actually shipped Release 12 for Windows on February 16, 1993, *before* Windows NT or Win32s were shipped. We used the WATCOM Windows expander which allowed us to run a 32 bit application under Windows 3.1.

JW: Really before that. We'd shifted most of our people onto Windows well before Information Letter 14.

RC: Well before that.

GZ: Ok, okay.

AG: Oh, that was '91, did you say '90?

GZ: Uh, when was Windows 3.0 introduced? May of 90, May of 90?

JW: 90, yeah.

GZ: Yeah, that's what I'm talkin' about, so it's around May of '90, I'm askin' 'ya whether prior to the introduction of 3.0 did ya do all this shifting of resources or was it, was it, after, yeah, cause this, this, this memo is a, is a whole eleven months after Windows shipped and by then it was, I'd say...

RC: We already had the resources working on...

AG: We were already working on...

GZ: *(Interrupting).* Yeah, I'm asking you when, though, I mean was it after May of '90 that you...

JW: *(Interrupting).* Well, Ruth wasn't in charge of that at the time. I was in the middle of the product at the time, so perhaps I should give you the facts. We were working with the 3.0 SDK before 3.0 ever shipped. We were a Windows developer receiving the SDK and putting the product on there as soon as anybody else...

GZ: *(Interrupting).* And was there a point in time where you decided, hey, let's really throw a lot more resources behind getting this Windows product done?

JW: Yes, and that—I don't recall the precise month—the month that the whole OS/2 blowup happened, when we realised that nobody...

GZ: *(Interrupting).* Oh, with IBM, okay, so January, so so January 91.

JW: No, no. That was some time earlier that it was clear that OS/2 2.0 wasn't going to come along...

GZ: *(Interrupting).* Oh, okay. All right, you're talk, okay, all right, all right, I thought, you, mid, mid, '90, okay.

JW: We work very closely with Microsoft. And what we were hearing from Microsoft during this entire period of time was that OS/2 was the engineering platform of choice. And it only became clear in mid-90 that it wasn't going to be, and that...

GZ: *(Interrupting).* But didn't it become clear to 'ya in mid-'90, just a few months after Windows shipped, then, you're saying. That's, that's mid-'90, I mean, May...

JW: Is this a Congressional investigation?

GZ: I'm just, it's, it's important, I'm just, it's important that I get some sense of accuracy about when it was.

JW: I don't spend all of my time writing down logs of what of what happened...

GZ: *(Interrupting).* Well, could somebody check on that, uh, uh, because, because...

JW: I don't understand why this is important. We were working...we were, as we always have, on every potential platform that may be a contender to be on the engineer's desk. And that means we were working on OS/2 1.0, the 2.0 prereleases, the Windows 3.0 SDK, all at the same time. In terms of the point that we said, "Look, OS/2 just isn't going to make it in the near future, let's move all those people over to Windows and stop working on OS/2," I don't recall what month that was, and I don't think that's...

GZ: *(Interrupting).* Well, all, all I'm saying is that in April of '91 you said "We have to revisit the level of support we're planning for Windows" and prior to that, page 13 you say, and it's, this is quite true, "this makes getting caught out without a Windows version of your program just about the worst possible thing that can happen to a DOS application vendor these days." That's true, and 'ya talked about how the, uh, "one central and virtually unquestioned tenet of Autodesk's strategy has been platform independence" and I thought you made some, um, astute, uh, analysis of the, of the problem with that and you said, "we have to revisit the level of support we're planning for Windows." So in April of '91 there's some, there's some sense here that, that *(guttural)* there, there wasn't enough support for, for, for Windows. Uh, that's, that's the only, that's the only thing that, that the English seems to indicate here.

JW: There is? There is? I...

GZ: *(Continuing, ignoring attempt to reply.)* "A similar failure to comply with the ground rules for Windows applications may hurt us severely...

JW: *(Interrupting).* Yes...

GZ: *(Continuing, ignoring attempt to reply.)* "and every week that passes without our thinking about how to address this problem adds to the danger." So it's clear that there was some danger in April of '91 that you weren't giving Windows enough attention.

JW: There is some "danger" in February of 1992 that we are not giving Silicon Graphics, or Hewlett-Packard, or any other contender...

GZ: *(Interrupting). (Condescendingly.)* Well, if you were exaggerating there, you can, you can tell me, you can say, "I just was exaggerating."

JW: Are you cross-examining me, or would you like me to answer in...

GZ: *(Interrupting). (Condescendingly, with a big grin.)* You can perceive my questions in any way you want. I'm just trying to, to use the materials at hand here.

JW: Sir, I would answer your questions a lot more if you would listen to my answers and let me finish and not interrupt every answer that I give and...

GZ: *(Interrupting).* If we had, eight, or ten, or three days, yeah, you know, but we have a couple hours, and we have to move the thing along. I'm sorry about that, but it's not me who put the time constraints on it.

JW: It's you who were late.[408] *(Pause.)* Let's go on then. I think, in...

GZ: *(Interrupting).* But that's the question, though, the question is, is...

[408]The interview was supposed to start at 3 P.M. We sat around, cooling our heels, until Zachary turned up 20 minutes late.

JW: *(Interrupting).* I have felt, ever since it became clear to me that OS/2 was not going to become the platform of choice, that there was a great urgency in getting onto Windows, because at that point there was no other possible contender on the horizon unless IBM suddenly came back with an OS/2, as now they may be, but that in any case Windows was going to be the first platform after DOS that dominated the desktop of our prospects. At that point, and at the point that I wrote that I believed that 32-bit Windows was much sooner to hit the market than it now is. I was talking about 32-bit Windows there, and I expected at that point 32-bit Windows would be shipping before the end of the year. If 32-bit Windows is a '93 project, I feel much less urgency than I...

GZ: Okay, okay, all right, fair enough, that explains, that's, that's, that's, a good explanation for it, uh, I, I, I mean the terminology and the shorthand often, prevents us from, you know...

JW: Well recognise, that memo was not written for you, it was written for people within Autodesk that understand...

GZ: *(Interrupting). (Shouts.)* Oh, I know, I, this is why I'm asking you questions about it. Because I'm tryin' to understand what you really meant. If I didn't really, if I didn't care about that, I, I would just take the words at the face value and move on. But I do care about trying to understand what you, what you meant in this, in a, in a, in a, in a passages. Uh, okay, all right. *(Long pause.)* One, one of, one of the things about the, about the, uh, analysts, and I talked to some analysts after the meet, I may talk to 'em all, but after a couple you got the flavour of it, uh, um, *(guttural)* hah, one, one of the things that was funny is that, I mean, here you try to explain to them, hey, we're trying to tell you something that we think's better for the long term of the company, you know, try to focus on that, don't try to get caught up in, you know, you have a tendency to be short-term, course the reaction was totally short term—people, people just felt, you know, ehhhh, the short term looked terrible to them, they're very, uhhh, I would say, ah, people were very disappointed, I mean, in, from the standpoint that, that, uh, uh, they didn't understand, they, they, they, they really didn't hear you, in, in that sense and, and, and I just, I, I, I, I think that, that, uh, you, you guys strike me as people I would, stick, stick to your guns and, and, and, and keep your eye on the ball but, but how do you, how do you manage these expectations of Wall Street and how, uh, is there, I mean, are you imper, can you be impervious to their wrath, or their dissatisfaction, I mean, is there just a way to say, well we're just going to put the blinders on and we're not going to care what they're going to say and just do our game plan, I mean...

CA: *(Interrupting).* I think that's what we've done, and we certainly haven't changed our strategy in the last week, coming back from the meeting. So, I think that...

GZ: *(Interrupting).* You, you didn't feel, you didn't all go, get a stiff drink after that meeting, or something, uh?

JW: I believe that if you talk to the investors you find many of the investors far more sophisticated to what high-tech companies have to do in times of recession than many of the analysts are.

GZ: Yeah, yeah, well that's, that's, that's, that's certainly, uh, do you have, what, big institutional, are you referring to big institutional investors.

CA: Yes. And we met, that evening, with a number of institutional investors as well as the entire next day.

GZ: *(Interrupting).* So separately...

JW: And in Boston as well as New York.

CA: We went to Boston on Friday.

GZ: And you talked with representatives from these, these are funds, some funds or another, uh...

CA: I think we calculated that within 24 hours of the meeting we had talked to something close to 50% of our shareholders.

GZ: Really?

CA: Face to face.

GZ: Wow. And the reaction from those guys was...

CA: For the most part very supportive.

GZ: Okay, cause the, the, uh, quite a few of them are long-term holders, long-term holders of, of your shares. Okay, okay, uh...

JW: And they didn't all necessary buy in at the top, either.

GZ: Hah, hah, hah, hah! Let's hope not! Right, *(unintelligible)*, but, um, okay, so, so you feel that, uh, you've been able to kind of poll some of these folks and, and you feel that you have, maybe the analysts have been queasy about all this but, but the institutional investors are, are, understand what you're trying to, uh...

CA: I don't know if poll is the right word. It isn't that we go out and ask for their advice; we've had the opportunity in a smaller group...

GZ: *(Interrupting).* No, I meant, I meant you told them after, you told them what you were doing, you, you got their, you took their pulse, yeah?

CA: Right. And in a small group we were able to, you know, continue to address their concerns.

JW: And I don't believe it's accurate to characterise all the analysts as negative. I've read a lot of the analysts' reports and it's one thing to say "we're worried about the next quarter of the next year" but many of the analysts do focus on the market share, on the strategy of the company, on the fact that we actually gained market share last year, and we're well positioned in this. I think the analyst is really serving two communities. There's what's going to happen next quarter vs. what's going to happen 12 months, 24 months out.

GZ: Yeah, no, that's true, you're right, yeah, you're right. Uh, one concen that came up with them was, was about this leadership question uh, and maybe we can, you know, I mean, is, is, who's leading the company and there's a concern about the, the, uh, interim status of, of, of John and, and, and, and, uh Volker and, and concern about who your replacement will be and, uh, what, uh, what, what, what can we, uh, can we, what, what can we expect with regards to, to that, uh...

AG: Well, I think as far as, you know, if there's any... and once again going back and quoting another, another newspaper, you know, you, which said "and we have the three lame ducks, Green, Walker, and Kleinn," I would say, if you look at what has happened in the company during the last few months, there's probably more activity coming out of this company than there has been for quite some time. So I would hardly consider what is happening the strategy of a group of lame ducks. At this point, once again, I'm the CEO of the company, Volker came over here in the newly-created position of Chief Operating Officer and, to

be quite frank, that's a position I should have created about a year ago. Believe me, that certainly...

GZ: *(Interrupting).* With him... with him in it, I mean, or, or, or, just, but just a position...

AG: Not necessarily with him in it, but, I mean, creating that position with someone in there. I mean, the amount of stress that has relieved just in the period of time he's been here has certainly made a...

GZ: *(Interrupting).* Does it make you want to stay?

AG: Well, I'm not going away—I'm not disappearing. I'm staying as Chairman of the company but, you know, I've been doing this for quite some time and, in fact, I've been talking about for the last two years of getting out of this on a day-to-day basis. If you want to imply anything from Information Letter 14, it probably prolonged my not getting out of here by that period of time until I felt that the company was...

JW: *(Interrupting).* Of the three lame ducks, Al is continuing to hold his position of Chairman for the foreseeable future, Volker is continuing to be Vice President of European Operations, which is the job that he wants, and I'm continuing to work on software development. So after we hire a new CEO, everybody will remain in a rôle that they currently have, interim changes notwithstanding.

GZ: Will you also have a new COO then, as well?

AG: That's really up to the CEO, but what I would do is when he comes in, I would highly recommend it to him. Before the reorganisation—I forget how many direct reports I had, but it was in the twenties. After the reorganisation, it only went down to about 18.

GZ: *(Interrupting).* In the twenties? You had direct reports in the twenties?

AG: That's the way the company evolved. Well, even after the reorganisation it only got down to about 18. Once you put a COO in there, which takes responsibility for the business units, and then essentially then you have...

GZ: *(Interrupting).* Okay, all the business units report to, to...

AG: ...to the COO, and then you have essentially the officers of the company reporting to the CEO. It looks very much like a structure should...I don't know what it should look like—but it feels more like a structure should feel like, from this end.

CA: Let me just say one more thing to that, and that is that, as far as the management team goes, the rest of us have been here anywhere from, if you look at some of our general managers are brand new, to myself who's been with the company for seven years. And so, even though there is some transition in some of those positions, I'm not going anywhere and I think the rest of the management team...

GZ: *(Interrupting).* There's a lot of stability.

JW: And all of our business unit managers, except those that have recently joined, were promoted from within, within the company. So there hasn't really been a tremendous shift in people; we just have a structure that works a lot better in terms of running a company of this size. We grew from almost from $50 million to $250 million without ever shuffling around the reorganisation, and it just accretes to a point where you have too many people to one person for them to be able to effectively manage.

GZ: Okay. Do, do you expect the CEO to be from within or without, or, uh...

AG: We have a search firm who's analysing this—they're looking on the outside—they came up with a list of some very qualified people in the industry that they thought they should contact. Once again, they've broken that down to a short list; they're now contacting those people. There's some good talent out there. So we will interview and see what happens from there, but right now we're trying to find the best possible candidate from a world-wide search.

GZ: Uhhh do, do you have any kind of time frame, or...

AG: Just based on a gut feel of the process so far, it looks as though probably somewhere in the, I would imagine, April-ish time frame, probably in a sense.

GZ: So that's not that far away, that we're talking about, okay. What uh, I mean, what, what advice will you give to someone about, um, the proper relationship to form with John Walker?

AG: *(Long pause.)* To a great degree, a person who comes in will probably have had relationships somewhat like that in other companies. I mean, in a lot of companies you have situations where you have founders and you have CEO's that come in that are not founders. If you look at the relationship between John and I—I've known John for eight years. I consider the relationship to be a good one. I can think of a lot of other relationships with other founders of other companies that would have been quite painful. So, to a great degree, I would expect that the person coming should have those interpersonal skills already in place. I will be on as Chairman of the Board and whatever coaching or assistance I can give that person I certainly will, but to a great degree I would hope they would possess those skills when they come into the company.

JW: And if I can talk about the other side of that relationship.... It's a little difficult to talk about a personal relationship with somebody that you haven't met. I will certainly say to the new CEO what I've said to Al on many occasions: I'm not interested in running this company. I, as a shareholder, am interested in this company being very successful and fulfilling its potential. If, I think, any time in the last six years, since Al had become president, he had said, "John, you're not helping. Just be a shareholder," I would have said, "That's fine with me. If you'd like me just to sit home and program, I'll sit home and program; if you'd like me to just be a shareholder, I'll be a shareholder." I have no interest whatsoever in pulling strings or being a hidden power. I have an interest in developing products that I believe are very important for this company. That's what I want to do, and nothing will make me happier than getting back on the plane to Switzerland and going off and disappearing for three years developing new products for this company, and I hope that, when the new CEO comes in and takes charge that he will have no need to interact with with me other than calling me on the phone every now and then or seeing demos of the new products I develop. I had that relationship with Al from '86...

GZ: *(Interrupting).* Is that, was the, that was that the relationship you had with Al from, what, '86 through, or, in...

JW: Through this week. I mean, we would speak, about two or three times a year for a couple of hours...

GZ: *(Interrupting).* On the phone, on the tel, tel, telephone?

JW: No, in person.

GZ: Person.

JW: And everything went just fine in that regard. I don't believe there really was any great conflict. It's not a conflict issue. It's an issue of a company that needed to change its fundamental mindset, from top to

bottom. Not Al's mindset, not Carolyn's mindset, not Malcolm's mindset, but everybody all over the company. And that's a process that companies have to every now and then if they want to survive when the rules are changing.

GZ: *(Interrupting).* Agreed.

JW: And that's the process that I'm hoping to contribute to here. I'm hoping to get it done as soon as possible.

GZ: Has it been painful for, for the organisation to go through this?

AG: In some parts, I would imagine there was pain associated with it, but the opposite side of that coin is just denying it and doing nothing about, and losing your competitive edge would be even more painful. So I think everybody realised it. It's one of those things that, "Oh Damn, I hate it when he's right," but the other option is totally to ignore it and I think everybody in the company looked at this and said, "Yes, there are things that need to be done." And rather than just denying that the letter was ever written or denying that we were in that position, everybody pulled together and I think the company is in a lot stronger position for it. And, in fact, if you want to look at being on the receiving end of that, I think it would Damn healthy for a lot of other companies in the United States to go through the same self-analysis.

GZ: I, I, I, you can make a long list, yeah.

JW: I have to say, ten years ago I held this meeting in my living room and 16 people came on a Saturday. Well, last weekend we held a meeting in the Marin Civic Centre, voluntary meeting, Saturday afternoon... we filled the Civic Centre with employees of the company...

GZ: Was that to, explain, uh...

JW: It was to explain what we going to try to accomplish in the next ten years...

GZ: This was coincident with the analysts' meetings? Is that, is that...

JW: Right after we got back from the analysts' meeting.

CA: But it was also...

AG: *(Interrupting).* With the beginning of the second decade of the company.

GZ: Well was it the tenth anniversary, this was the actual tenth anniversary?

CA: A few days after.

GZ: Oh wow. So it was kind of a bittersweet party, or, or, *(chuckles)*...

JW: I wouldn't describe it as bittersweet at all. I mean, the first thing there were some cynics saying that we might get turnout of fifty people, and we literally filled the Civic Centre.

GZ: Which was, what, several thousand?

CA: We don't have that many people.

GZ: Sixteen hundred...

JW: Six to eight hundred, we only have seven hundred employees in the U.S. But we had essentially complete turn-out of all Bay Area employees voluntarily came to this thing, and I think that really indicates the spirit. I mean, this was a voluntary meeting, primarily to hear the senior management and the business unit management describe what we're going to try to accomplish in the next year and the next ten, and everybody showed up, startling some people who didn't think they would.

GZ: Did you give an address?

JW: Yes.

GZ: So, and you did and, and they all, okay, so, so you made presentations.

JW: And we demoed all the new products and talked about them.

RC: And it was two o'clock on a sunny Saturday afternoon in Marin...

GZ: *(Interrupting).* When you started, or, yeah. When did it, when did it end?

LG: Five, five-thirty.

GZ: Five, five, that's fairly efficient in these, hah, hah, they are talking...

??: We got kicked out.

GZ: You got kicked out. *(Laughter.)* Yeah, okay, okay. Well, I mean do you see, I mean, there's obviously been a therapeutic, um, effect from this Information Letter and, as you mentioned, I guess there was no, no 13 but there were twelve before that. Will, will there be a 15 at some point?

JW: Well, there was no 9, actually.

GZ: There was no 9, either?

JW: There was a 12.

GZ: There was a 12...

JW: There was a 12 and 13, but I just missed 9, and...

GZ: Oh, oh, okay, all right, sorry then.

JW: If you read my book, *The Autodesk File*, there's a story about one where 9 disappeared.

GZ: Oh, okay.

JW: No, the...I, I will write one if I feel the need. Right now, I think that Information Letter 14 stands, in my opinion, as an accurate portrayal of the transitions we're going through right now and probably—I don't see at this point, unless some huge surprise happens like Microsoft withdraws Windows or something, that the competitive environment is going to change very much. It didn't change very much between '82 and '90, really, in terms of our market. And, I don't expect to see any kind of really radical shifts beyond the kinds that I discussed there. If I do, I'll certainly bring them to the attention of people. I hope that as Autodesk becomes a more reactive, rapidly moving, rapid time to market company—the changes we've made over the last year are all to that end—that we will be able to adapt more smoothly to the changes in the competitive environment, rather than with a big lurch as was necessary this time.

GZ: But you, you, but you see, you, you, your, yourself returning to, the, the programming work you want and, and in terms of the follow-through on the implementation of all these things, you're, won't be involved in that.

JW: I don't see any need to be. I believe that we have strong managers who, with the support of a CEO that understands the opportunity we have, can bring the products to market. I'm not an expert in developing marketing plans or putting together roll-out plans or in public relations. We have people who are, and they're the ones who have to do that. If I'm an expert at anything, it's finding opportunities that we should be getting into that are going to pay off in four or five years out, and that's something that's best done with a very small group of people with very limited funds, and then brought to the attention of the company when that opportunity begins to manifest itself.

GZ: What, what, what are some of the programming accomplishments that you're, you're most proud of, or you think had the biggest benefit for Autodesk in, you know, over, uh, in the second half of the, of the Eighties rather than the whole course of the company, but what, what are some of the things that in recent times that, you, you felt had the, had the...

JW: I think making the prototype of AME, demonstrating that we could integrate solid modeling with AutoCAD, is the one that springs to mind as the most important for the company. That's really what let us position ourselves as something a lot different from a drafting company. Nobody else had ever, essentially, included solid modeling as an integral part of a CAD system.

GZ: When did that happen?

JW: The prototype was completed on July 20th, 1989.

GZ: Okay. And and, then, uh, is, is that typical, I mean, and then, then, some, cause someone has told me this story, then you, did, did you, did you hand, that, do you then hand that off to, uh...

JW: Yeah, I typically hand that prototype off to a product division... well, the prototype is presented. People say "Is this a worthwhile thing to do?" That happened to be one that was accepted very quickly. I've done others that never have been accepted, and that's fine too. But then a software development process begins. That was a complicated one because we were selling a stand-alone product as a solid modeler at the time at at a $5000 price point, and I was proposing integrating it with AutoCAD and selling it for $500. So that's not the kind of decision you just throw up the coin and make instantly, but the decision was made to do that. A major software development effort that lasted the better part of a year began, and involved close to a dozen people...

GZ: *(Interrupting).* Were you involved in that?

JW: No. I was not involved in that. I was off...

GZ: *(Interrupting).* So you handed off that prototype and then...

JW: *(Interrupting).* And went on to work on the next project.

GZ: Okay. Now, so, is that the, your typical style, to do, to to, bring a, a, a, a, a, a, a, a program to that point where you can then hand it off to a development team?

JW: No, I have also was...

GZ: *(Interrupting).* Okay. You will finish a product, too...

JW: I did the product after that, CA Lab, entirely myself with the help of one other person, all the way to shipment.[409]

GZ: Oh, okay.

JW: I did...

GZ: *(Interrupting).* Wait, I'm, I'm, I'm sorry, I've got to catch up with you, you. *(Pause.)* CA Lab was done after, or before AME.

JW: After AME.

GZ: After AME. And when did you finish that one, do you recall?

JW: I don't recall it, I'm sorry.

GZ: That would'a been '91, though, or, '90, or...

JW: That was, probably late '89.[410]

GZ: Oh late '89, oh, okay, okay, so shortly, on the heels of...

RC: *(Interrupting).* Release 12.

JW: Well then, in another mode, I was involved in the down and dirty debugging of Release 11. I often jump into debugging the final phase of AutoCAD and proofreading manuals because I've written lots of pieces of it and I have knowledge that kind of spreads over the product. In general, if Ruth or somebody says, "John, we need you to help on this," and convinces me that's the best way I cen benefit the company, I'll jump in and do it.

GZ: How do you usually broach the subject with him, E-mail, or, uh...

RC: E-mail, or just ask.

GZ: Just, what, call him up and ask him, I mean...

RC: Just ask. You can always find him by E-mail.

GZ: Yeah, yeah. Uh, okay, okay, so, so, you, you, you got a variety of experiences here between just doing a, doing a product like CA Lab from, from start to finish, and...

JW: With one other person.

GZ: With one other person, okay. Was that person in Switzerland with you, or, or...

JW: No, no, I was here in California.

---

[409] Reading this over, it occurs to me that this could be taken as denigrating Rudy Rucker's contribution to CA Lab. That wasn't intent—I'd say we each did about half-and-half of that product. I made a point during the interview to never name names unnecessarily, for fear Zachary would get on their case as well as mine.

[410] June 1989.

GZ: Oh, you were in California at the time for CA Lab? Oh, okay. When did you move?

JW: May of last year.

GZ: May of last year, okay. Sorry. Uh, well so you moved shortly after, after, after uh this memo.

JW: Yeah. I had announced that I was going to move before the end of 1990...

GZ: *(Interrupting).* Oh, okay. And prior to that, prior to that you were, you were living here. You were living in, living in California. Okay, uh so that's why you saw more of him, then.

JW: No, I would not say saw more of me. My communication was primarily E-mail or telephone...

GZ: *(Interrupting).* Even when you lived here.

JW: ...even when I was living in Muir Beach. I think I actually, for the last 6 months, I've gotten to the U.S. more than I got from Sausalito from Muir Beach in '89.

GZ: *(Laughs.)* Um, so, so even something like debugging, now what, uh, AutoCAD 12, when was that rel-, that was, that was the, that's the current release of AutoCAD.

RC: That, we're working on that will be shipped in the second quarter of this year.

GZ: And you, you actually worked on debugging of, of that, that...

RC: John...

JW: Debugging and development of some pieces of it, yes.

RC: But he has a major feature that, that John implemented within this release.

JW: Well, and I think in terms of continuity, I was working on that feature before I got on the plane to Switzerland. I got off the plane and went on working on it and submitted it from Switzerland. So in terms of what I was doing, moving to Switzerland made absolutely no difference at all.

GZ: Is there, is there anything cumbersome about the physical separation?

RC: No.

JW: We've been that way since '82. Many of our programmers still work from home...

GZ: *(Interrupting).* Oh really...

JW: ...and we're all linked together by fast data links and E-mail and, in fact, in Switzerland I'm a faster data link than I was in Muir Beach. So...

GZ: You, it, does it actually come faster?[411]

JW: It's...

GZ: *(Interrupting).* It's close.

[411] Yes, but never prematurely.

JW: Well, it was 9600 baud, now it's 56 kb, so...

GZ: That's, that's, hah, uh. And you work from home?

JW: Often, although I work more out of the office...

GZ: *(Interrupting).* Oh there is an office, though? There is an office there. Um... *(Pause.)*

JW: It's not just a technical centre. We're actually going to be doing our manufacturing and shipping from there. We're centralising...

GZ: *(Interrupting).* This is—how do you pronounce the name of the town?

JW: The canton is Neuchâtel. We're in the town of Marin, M-A-R-I-N.

GZ: M-A-R-I-N—that couldn't have been a coincidence.

JW: It was entirely a coincidence.

GZ: So, so a lot of your programmers do, do work at home, uh, so, uh, within the AutoCAD project, uh, the different releases, you, you, you, now you have a, an R&D manager that reports to you, right, or...

RC: I have a counterpart who's a Chief Technical Officer.[412]

GZ: Ah, okay, that's a chief, okay, so the chief technical officer, he's probably got the AutoCAD project, uh, feature list, broken up into different groups, and then each group member reports to 'em or something. They must get together and see each other, but the members of the, of those groups, some of them may be working at home.

JW: Yeah. They'll typically come in once a week, or...

GZ: Okay, okay. Okay, a lot of people talk, I mean, a lot of people talk about this place having, you know, this culture of the programmer, and the, I mean, whu, how does that evidence itself, just, just in, in, uh, I mean, I, some people said to me they let you bring your dog to work, uh, I, I, I, I don't exactly consider that the litmus test of a, a great employer, but, uh...

RC: The AutoCAD business unit's finance manager brings his dog and parrot to work, so I don't think it's unique to programmers.

CA: Not to mention the credit manager brings her dog to work.

JW: And that was the first dog.

CA: And she was the first dog. It started in accounting.

JW: In terms of the programmer culture, I've never understood this "programmer culture" *per se*, in the sense that within the first eight months of the company, essentially right after we started shipping our first product, we immediately had more people in marketing and sales that we had programming, and have ever since. In the mid-'80s, when we were so successful and nobody was writing stories about is, the word on the Street was that, well, we were a company with mediocre product and great marketing, and...

[412] John Lynch.

GZ: *(Interrupting). (Laughs.)*

JW: . . . and that was the common slam that our competitors would throw against us, okay? And so, I mean, that we have always been a very aggressive marketing/sales/distribution company. We've built a dealer channel that didn't exist, and we control that channel, and it's a great asset of this company. You don't succeed at that if you have a mediocre product. You gotta have a good product. But if I look at this company, I don't see it as being unbalanced in those regards. . .

GZ: *(Interrupting).* Well, I wasn't suggesting it's unbalanced, but I'm just saying there seems to be a lot of, a fair amount, of, I mean, it's unusual for the kind of candid debate that you, you have had. I mean, you, you, you mentioned, you know, there are other companies, there certainly plenty of other companies that are ripe for it, more in the older industrial sectors of the economy, but, I mean, a lot of these companies are very hierarchical, I mean, you gotta remember, we don't just, uh, our readers just aren't in California, I mean, you know, you gotta, you know, when I had this conversation with you the first time on the phone,[413] I, I tried to impress upon you that maybe you took it as a matter of course that you might have a candid discussion with John about, you know, how things were going, but in a lot of companies, it's considered very unusual, you know, the, the wisdom comes from the top and it goes down.

AG: But I think the unusual part of the company is that the type of—the reason that attracted the people here in the beginning was probably that structure. . .

GZ: *(Interrupting).* Well, that, that, that, I'm asking 'ya, I'm tryin' to. . .

CA: *(Interrupting).* Can I jump in on this?

GZ: Yeah.

CA: That's as much Al's style as it is John's. I mean, that's a credit to Al's management style and, I would guess, John, not putting words in your mouth when you were talking about Al earlier, it's exactly because Al doesn't come in and say "I know everything and therefore you're going to do it my way," that the organisation has developed the way it has. And so that's as much Al's style, you know, as much as anything else.

JW: This isn't a matter of some, you know, California life style company or something like that. We do things that are effective and productive. And if a programmer works at home and works 16 hours a day—if he'd had to spend two hours commuting to the office, that's a productivity gain. If it weren't a productivity gain, we wouldn't do it that way. Many of the things that other companies are talking about—telecommuting, well, that's kind of exotic, I guess it's becoming pretty commonplace now—well we've been doing it since 1982. And it was *tough* in 1982. . . you know, like mailing floppy discs around, okay, but. . .

GZ: *(Interrupting).* Were you mailing floppy discs. . .

JW: We were mailing floppy discs.

GZ: Through the U.S. Postal Service, you're talking about. . .

JW: Through the U.S. Postal Service, that's right, that's right. We were mailing the AutoCAD source code around and we did that until, well '84 was when we got the first E-mail that worked.

[413] Speaking to Al Green.

GZ: Hah. That's funny....

RC: Also, our programming staff has very good business sense. They will ask, "Why do you want this particular functionality or this feature set in a product?," and what is the market case for it and are we, in fact—is this going to be the quote-unquote "twenty-five million dollar feature." So, before they program, they're very. vary careful and clear about wanting to know what this is going to do to us and for us in the market place.

JW: And in terms of who we recruit, if you look at our employees throughout the company, our technical people are not, for the most part, mathematicians, super experts in R&D—many of them came out of AutoCAD dealer environments. Many of them were customers who started using...

GZ: *(Interrupting).* Your programmers...

JW: ...and, and became programmers, that's right...

GZ: *(Interrupting).* That's, that's unusual.

JW: So we've tried to bring people—and on the sales side at least as much—throughout the whole...

GZ: *(Interrupting).* Many, of your, your, many of your programmers come out of, come out of the user community or, the dealer community...

JW: That's right. Because we've encouraged our dealers to become software developers. Those developers develop products—there's 400 products that work with AutoCAD...

GZ: *(Interrupting).* How many programmers would you say ya' have at the company then, uh, I mean world, world wide, what would the world wide number be, I mean, we're talking 500, uh...

RC: No, no, no, no, no. Two fifty—three hundred.

GZ: Two fifty.

JW: No. It's about 150. 165 was the number I used to use last year, but that doesn't count some people overseas.

GZ: So less than 200.

RC: Less than 200 is accurate.

GZ: Huh. Whu, whu, are you, um, trained as uh, double-E or computer science or, uh...

JW: General engineering.

GZ: General engineering. What, you have a BA or, uh...

JW: BS.

GZ: Are advanced degrees common here, or...

JW: We have people with them; we have people without them.

GZ: Yeah. Okay. Hmmm. Uh.

RC: Common sense is probably more preferable.

GZ: Are you an engineer?

RC: No.

GZ: No. You're not. Okay. What's, what's your background?

RC: Eleven years with General Electric in manufacturing.

GZ: Uh huh. Back East?

RC: Back East—southern California.

GZ: When did you join?

RC: 1988.

GZ: Oh, eighty-eight, okay. Uh, did, have you over the last year or so increased the number of programmers you have in response to, is that, that was one of the things, that, that, that came up was the, the, the, need to put more resources into, into creating software, is that, have you, have you, is there anything specific there that about how many you've added or, or is it, uh...

RC: We've added about sixty people to the AutoCAD business unit and...

GZ: *(Interrupting).* Sixty programmers?

RC: No about half of them have been programmers.

GZ: In the past year, this is, or...

RC: In the past six months.

GZ: Okay, added thirty...

RC: *(Interrupting).* And if you know of any highly-qualified programmers, I'd love to talk to them.

GZ: Hah, hah. Why, you're still looking for more, huh.

RC: Yes, yes.

GZ: And now why have you done that?

RC: Because we felt that we really need to go back in and invest in the core technology and invest back into AutoCAD and the modeling extension and other sets of products that we know that we're going to be bringing to market over this year and next year.

GZ: Uh, okay, so when you say invest, then what, what you mean, you need to have more people working on the basic stuff...

RC: On the basic product, and on additional products sets that we plan to bring to market.

GZ: Okay, and so the effect will be, you'll have faster time to market plus richer products, or...

RC: Most definitely.

GZ: But both of those, is where you're going.

JW: And understand, when we talk about a one product company, the AutoCAD business unit can be a fifteen product company in five years all by itself, and many of those products can be generating revenue on the AutoCAD scale, in terms of going into vertical markets, addressing specific customer profiles. So we're diversifying both in the other business units and within the AutoCAD business unit and many of those additional people are working on new products within the AutoCAD business unit. One of which is announced and will be shipped in fourth quarter—the constraint manager.

GZ: Okay. *(Long pause.)*

LG: We're getting about to the end of our time, but one of the things that to an earlier question, I'm not sure if, *(unintelligible)*, you were starting to talk about the market opportunity for Autodesk since early on, you know. Do you think you can elaborate on that?

GZ: Okay, sure.

JW: Well, I think there's kind of the, what I call the short-term, like two or three year opportunity and the long-term opportunity. The long-term opportunity is that this company can sell software into every part of the manufacturing process, of every product that is made over the next thirty years. And that's what we're working on—that is the opportunity. That is something that isn't going to go away this quarter or next quarter; that is something that none of the mainframe CAD companies that are currently addressing very, very small very, very expensive markets are targeting. That's something that even a retail drafting product that came out cannot address. The big picture is that around the year 2020 or 2030 there'll be 10 billion people on this planet, not 5 billion, and those 5 billion people will be using products which means that in the next 20 to 25 years, on this planet, we're going to manufacture as much stuff as has ever been manufactured up until now. And it's out goal to be participants in every part of that manufacturing process. That's what this company's trying to do, and that's not something that 'ya do next quarter or the quarter after, but it's something that each one of those products that comes out fills in a piece of that puzzle, so that in ten years you have a factory which is totally automated, from conceptual design to manufacturing, and each component of software in that chain is a component of software that comes from Autodesk—whether it's something we've written ourselves, or something we're selling, or something we've acquired the technology—that's not the point. The point is that we have a solution for the entire manufacturing process, for the entire architectural design process, for the facility management process, for the mapping process. Each one of those markets is an enormous market of which drafting is one little piece of the puzzle...

GZ: *(Interrupting).* Which is, which is, your, your core right now...

JW: Our core is drafting and that is the single most lucrative piece if everything stays the way it stays now. I do not believe that in 15 years there will be people running milling machines all over the world, the way there are now. That's going to become automated, and in the largest manufacturers it already has, but that's going to spread out throughout the entire economy around the world.

GZ: So you see a rôle for your software in that automation process...

JW: Without doubt. I don't see anybody else even working on it right now.

RC: And in different markets, too. Not just manufacturing. John's statement about facilities, about architec-

tural, about geographic information systems, about mapping is really what we're all about.

GZ: Okay, okay. Well one thing that would be good for me is I'd, I'd like to be able to, to talk to, uh, some folks within, sort of, the greater Autodesk community like some dealers and some, uh, uh, either large or typical customers, uh, to get some sense as to how people put your products to work for them and also to, get a, with the dealers, to get just a better understanding of this dealer network and, and, uh, that should give me a little bit better idea as to, as to, uh, uh, the comp, you know, the company's viewed, uh, but, and then at the end of the thing, it'd be nice to know that there was somebody I can, I can call with some followup question about some of the things that I've learned so, so, uh, I'd, I'd appreciate it if some can, some people can make some more time on that but, uh, thanks for, uh, getting together on this thing, and uh, uh, it's, uh, I'm glad, uh, I was able to get you guys and it sounds like a very uh, interesting, interesting situation and, *(sarcastically)* huh, I'll try to find out, find out more, find more it.

RC: Thanks.

AG: Okay.

Exeunt omnes.

Anatomy of a smear

Zachary's "profile" of Autodesk appeared on the front page of The Wall Street Journal on May 28, 1992—more than four months after the interview which had to be scheduled in such haste due to his "deadline." It was, as expected, classic Zachary slash-and-burn journalism, directed primarily against myself. Extremely little material from the interview was used in the article; it didn't fit very well into the "cabal" fantasy Zachary concocted. Below, I'll analyse some of the key misstatements in the article. I quote brief passages from this copyrighted article for critical purposes under the doctrine of fair use. Quoted material appears in *italics*.

Tech Shop

'Theocracy of Hackers'
Rules Autodesk, Inc.,
A Strangely Run Firm

Can the Latest CEO Survive
A Cabal of Programmers
Who Send 'Flame Mail'?

A Most Unusual Interview

by G. Pascal Zachary
Staff Reporter of **The Wall Street Journal.**

> *. . . Autodesk's founding genius, John Walker, a reclusive programmer who doesn't allow the company to distribute his picture or publish it in its annual report. In a rare interview granted for this article, a prickly Mr. Walker insisted that a reporter sit in front of a video camera, declared that Autodesk claimed a copyright on the ensuing discussion and debated the meaning of each question.*

Autodesk used to have my picture on file and supplied it to the press upon request. When I discovered, in 1990, having lost 70 pounds in the interim, that Autodesk was sending out pictures of the 215 pound Walker of yore, I requested that they return the wide-angle versions to me. . . I'm so vain. . . . Afterward, nobody ever asked me for a new picture. We never printed pictures of any executive or director in annual reports prior to the arrival of Carol Bartz. My feeling, shared by Al Green, was that it encouraged a cult of CEO personality rather than focusing on the company and its products, people, performance, and promise, in which the CEO is one of many contributors. Note the phrasing "a reporter," which dodges the fact that it was Zachary who agreed to be taped—if he'd said "this reporter," as you'd expect, he'd have acknowledged a precedent for future interviews. Read the interview and decide for yourself if I "debated the meaning of each question," as opposed to dodging the many verbal traps in Zachary's phrasing.

> *Unlike Mr. Gates, Mr. Walker, 42, never really wanted to run his company.*

Untrue. I ran Autodesk from its inception through 1986, leading it through our Initial Public Offering in 1985. In 1986 I decided that the size of the company demanded a CEO with professional management and financial skills that I didn't possess. Had I not done so, of course, the story would have been "Clueless Programmer Destroys Promising Company–'We Don't Need No Steenkin' Managers.'" And he got my age wrong, too—never mind.

> *But the real power still rested with Mr. Walker, Autodesk's biggest shareholder, and an elite group of programmers called "Core," who had either helped Mr. Walker found the company in 1982 or led its most important projects.*
>
> *Core members are contentious, eccentric, free-thinkers who have had a way of devouring professional managers.*

This fairy castle of utter fantasy has been repeated so many times that otherwise rational people are beginning to believe it. There is not, and never has been a group, cabal or otherwise, called "Core." "Core," around Autodesk, refers to the central components that made up AutoCAD—its guts, as it were, as opposed to device drivers, applications, documentation and tutorials, and suchlike. The group working on this "core code" numbered 10 as of mid-1991—among more than 700 domestic employees of Autodesk—and included only 3 founders, one of them half-time. It was managed by, and had been since 1985, professional technical managers drawn from outside the company or promoted from other areas within Autodesk. By the time of Zachary's interview, the majority of members of the "core code development group" were not only non-founders, but recent hires.

The founders who worked in this group, Duff Kurland, Dan Drake, and Greg Lutz, notwithstanding the latter two serving on the Board of Directors, were utterly uninterested and uninvolved in Autodesk politics and had, on numerous occasions, declined opportunities to participate in Autodesk senior management. Dan Drake had, in fact, retired as Executive Vice President in 1989, and announced his retirement from the board of directors 18 days after the Zachary interview. Hardly the actions of a power-mad "cabal."

As for having "a way of devouring professional managers," one must ask just which professional managers were devoured? Al Green served as Autodesk's president longer than alleged chief-cabal-conspirator John Walker. Every change in the senior management ranks I can recall in the years from 1986 through 1992, and there were relatively few, was made by the CEO, Al Green, based entirely upon his judgement. Other than being asked, on occasion, whether I agreed with the proposed change (and I always concurred), neither I nor any other founder or other old-timer was involved in any process of "devouring."

Finally, I wasn't Autodesk's "biggest shareholder," and hadn't been for years. The simplest cub-reporter check of Autodesk's proxy material would have confirmed this.

> *A year ago, Mr. Walker issued the ultimate in flame mail, a 44-page letter brutally attacking Mr. Green for allegedly trying to bolster short-term profits by neglecting investment in new products and marketing.*

I challenge you to find one place in Information Letter 14 where I attacked Al Green, brutally or otherwise. In Zachary's world, every issue facing a company boils down to a conflict between the egos of individuals—in this case the estranged founder, working through a "cabal," undermining the legitimate management of the company. Corporate strategy must flow from the head of a Superman CEO, all-knowing and all powerful, rather than drawing on all the intellectual resources of the company. The strategies I was attacking in Information Letter 14 were, for the most part, strategies I helped put into place myself in the early 80's. So, I suppose I could be said to be "brutally attacking" myself at least as much as Al Green.

> *. . . I do not believe the best decision is a group grope."*
>
> *That, however, is largely how Autodesk has been managed until now. It was founded by Mr. Walker and a dozen programmer pals. . .*

One wonders what phrase would have replaced "programmer pals" had Autodesk's founders consisted of real estate speculators, junk bond peddlers, savings and loan cowboys, and others deemed "legitimate businessmen" by The Wall Street Journal. In reality, our founders included a marketing and sales person with more than 20 years experience in the computer industry (Mike Ford), an investment banker (Jack Stuppin), and a prominent San Francisco corporate lawyer (Bob Tufts). Shortly thereafter we added a retired U.S. Army colonel, John Kern, to manage manufacturing and shipping.

> *Instead, Autodesk's hit product proved to be a computer-aided-design program that Mr. Walker purchased from an outside programmer named Michael Riddle. The program, which became AutoCad...*

This is a bald-faced lie, which the most cursory reading of *The Autodesk File* will demonstrate to be untrue. Mike Riddle was a *founder* of Autodesk, not an "outside programmer," and contributed the source code for INTERACT, with which Dan Drake and I had been working since 1979, to Autodesk in return for a royalty deal, just as Dan and I contributed major components of the Marinchip source code. Note that Zachary misspelled "AutoCAD" throughout the article.

> *Mr. Walker has unusual interests, which he imposed on Autodesk. When he grew intrigued with outer space, Autodesk invested in a company that salvages used fuel tanks from the Space Shuttle with the idea of sending them back into orbit, carrying the concept of recycling about as far as it can go.*

This is a lie, delivered with a nasty spin aimed at both Autodesk and myself. In 1987, I introduced Dr. Randolph Ware of External Tanks Corporation to Autodesk to explore whether an Autodesk investment in External Tanks could be beneficial to both companies. My sense was that the investment could be justified simply by the publicity Autodesk could derive from introducing AutoCAD into the very highest of high-tech domains, the Space Shuttle Program. That, and identifying AutoCAD's price-performance advantage with the cheap road to a space station that ETCO promised could easily yield visibility much greater than an advertising expenditure equal to the $225,000 investment sought by ETCO. In providing this introduction, I was simply putting the parties in touch, just as many other people did at Autodesk before and since—Autodesk receives dozens of co-promotion and partnership proposals every year, most of which it rejects, some of which it takes advantage of. As it happens, when Dr. Ware came to Autodesk to make his presentation to senior management, I was on vacation. Al Green asked me if I'd like to attend the meeting and I said, "No—it's your call." I heard nothing more about ETCO until, a month or so later when I returned, Al told me that we'd "written the check." Now, if this is "imposing my interests," I must be endowed with paranormal powers of persuasion—Al and other senior managers attended the meeting, they made their decision, and they made the investment. Other than setting up the meeting in the first place, I played no rôle in it. Shortly thereafter, we received a proposal to sponsor an NHRA dragster which had been designed with AutoCAD. I thought that was a cool idea too (though pricey), but it was rejected. Zachary's description of ETCO's business is totally wrong as well (see page 396), but that isn't germane to the slam inherent in the statement, just sloppy reporting.

> *He published a book containing scores of confidential Autodesk memos, many written by himself.*

This is a damaging, demonstrably false lie. The book Zachary is referring to is the Third Edition of the book you're reading now, *The Autodesk File*, which was published in 1989 by New Riders Publishing, a company in which, at the time, Autodesk owned a 1/3 interest. The copyright page reveals that it is Autodesk who holds the copyright on this book, not I. In fact, neither I nor Autodesk received royalties from it. The contents of the New

Riders edition of *The Autodesk File* were derived from the Second Edition, which was made available within the company to all employees, and was often given to prospective hires interested in "where the company came from." Before publication of the New Riders edition, the text was reviewed by Autodesk's legal and accounting departments, who suggested some minor deletions of material not considered to be public information, such as AutoCAD unit sales by month, and profit and loss broken out by subsidiary. All of the requested matter was elided, and *The Autodesk File* was published with the full approval of Autodesk's management, who considered nothing within it remotely "confidential," especially as all of this material and more were routinely made available to all employees. Other than handing over a copy of the disc containing the source documents to New Riders, I played no part whatsoever in the production and publication of this book, and nobody at Autodesk remotely considered it as "airing Autodesk's dirty laundry" in any fashion.

> *He is prone to making unexpected pronouncements. In a rare public appearance in March, Mr. Walker interrupted the description of a new product with this observation: "We are living on a small blue sphere in an endless black void."*

Zachary is referring to my talk at the introduction of AutoCAD for Windows in San Francisco on March 10, 1992, one of seven public appearances and press interviews I did in the three months I was in the U.S. in 1992. He was in the audience then, and despite spending most of the presentation talking out loud to one of his press cronies, when I wound up my talk with the message of the ultimate destiny of CAD and its place in the human future over the next several decades, Zachary raised his head and fixed me with a stare I will never forget. I've seen that kind of hate before—I've watched German newsreels from the 1930's and 40's—but I'd never before been on the receiving end. Read the talk he's referring to—you'll find it on page 781—and see if the phrase Zachary quotes was an "interruption" or diversion from the message I was conveying. Here was a software entrepreneur talking about technology and its place in the human adventure—an individual as yet unsullied by the tawdry greed, ego, and personality conflicts which consume so much ink in the daily press. I almost felt the targeting computer lock on—no need for The Force—"trust The Smear, Greg." A day or so afterward, our P.R. firm told me that Zachary has totally changed his schedule, or some such, and that he would be doing a much more "in-depth" profile of the company. I immediately knew what that meant. It was just a matter of waiting to see how bad it was.

> *These fits of impatience dovetailed with Mr. Walker's continuing suspicion of professional managers, shared by other members of Core. In early 1986, he forced out John G. Ford, Jr.…*

John G. (Mike) Ford was not a "professional manager," but rather a founder of Autodesk, as even the most cursory reading of *The Autodesk File* will document. As companies grow, things change, and sometimes changes have to be made. I have not, and I will never discuss the issues that led to this or that person's leaving Autodesk—in most such cases there's plenty of blame on both sides, and lots of shared regret afterward. But suffice it to say that when Mike Ford resigned in February of 1986, it was the unanimous opinion of the other directors and senior managers that it was in the best interests of the company. His successor, Tony Monaco was a 20+ year veteran of IBM.

> *Mr. Green was ill-suited to ride herd on the rambunctious Core.*

Which is why, one presumes, he managed, from 1986 through 1992, to lead Autodesk from $50 million to more than $250 million in sales, over four releases of AutoCAD created by the so-called "rambunctious Core,"

and was named, in February 1990, by *California Business*, one of the top 25 CEOs of the decade, receiving an award presented by former U.S. President Ronald Reagan.

> *Writing from his new home in Neuchatel, Switzerland, where he had recently moved to find more seclusion.*

As was made clear in the interview, I did not move to Neuchâtel until May of 1991—well after Information Letter 14 was circulated. As to having "moved to find more seclusion," I'd ask whether, a few months after my arrival, giving a 25 minute speech, in French, in the parliament chamber, before members of the government, business leaders, and a broad selection of European press constitutes "seclusion." (See page 709.)

> *Moreover, the broadside didn't mention that Mr. Walker himself had picked Mr. Green as his successor.*

Well, duh. Here is what I said in Information Letter 14.

> First a few words about me and my relationship to the company. As you probably know, I initiated the organisation of Autodesk, was president of the company from its inception through 1986, and chairman until 1988. Since I relinquished the rôle of chairman, I have had no involvement whatsoever in the general management of the company. ...Over the years I have agreed with many of their choices and disagreed with some, but all in all I felt our company was in good hands. In any case, I never doubted our senior management was doing a better job of running the company than *I* ever did when I was involved more directly.

Yes, I suppose a lawyer could argue that Zachary's statement is true, "the broadside" didn't mention that I picked Al Green as my successor, but seeing as I was Chairman of the Board at the time, and remained so until 1988, it kind of goes without saying that I played a major rôle in selecting Al Green. The "broadside" also failed to mention other dirty secrets such as the facts that water is wet, eggs break if you drop them, and that you can't always believe what you read in The Wall Street Journal.

Separated at Birth?

AutoCAD for Windows

Developing a version of AutoCAD for Microsoft Windows had been a key technical priority at Autodesk ever since it became clear that OS/2 was going nowhere (at least in its original 16 bit incarnation—at this writing, OS/2 2.0 remains a contender). I first tried to raise the priority of the Windows project in my "Max Q" memo in September of 1990 (see page 595) and, of course, throughout Information Letter 14. Shoehorning a huge program like AutoCAD into the 16-bit architecture of Windows 3.0 was a Herculean task, made easier by our initial investment in the OS/2 version of AutoCAD. Months of difficult work by the Windows team finally culminated in our announcement of the AutoCAD Extension for Windows in March of 1992. For only $99, any existing AutoCAD user could obtain the Windows version, allowing him to run AutoCAD both on Windows and directly on the 386 DOS machine.

We held a press conference at a hotel in San Francisco to announce AutoCAD for Windows. In my rôle as acting Manager of Technology, I was invited to "say a few words."

Remarks for the Windows Press Conference

March 10th, 1992

by John Walker

I'm John Walker. Thank you all for joining us for this announcement. Before we get into the good stuff—the details of the Windows AutoCAD we're announcing today, I'd like to briefly put the product into perspective, drawing upon the history of Autodesk and the microcomputer market over the last decade, the situation in the industry today, and the events we can expect to observe over the next several years.

I usually don't like to recycle material from earlier talks, but the breakneck pace of events at Autodesk in early 1992 left me no alternative but to reuse the sections of the talk I wrote for the special shareholders' meeting explaining Autodesk's industry position and technological strategy. Besides, the audience at a rollout of AutoCAD for Windows had virtually no overlap with the shareholders' meeting, and they needed to hear the same things anyway. In any case, the next several paragraphs of my talk were virtually the same as those at the shareholders' meeting starting at the paragraph that begins "I would like to briefly explain…" through "So now what?" Please refer to page 722 if you'd like to reread that material.

And next?

We must, at the end of this current technological transition, emerge with the same or greater market share for CAD on the new standard platform as we currently command in the DOS market. If we achieve this goal,

Autodesk's success in the next decade will be assured. As the current technological transition matures, we will enter an era in which the easily-drawn distinctions among "PCs," "workstations," and even "mainframes" begin to disappear. There will be, instead, a continuum of computing capability and cost that ranges from pocket pen-based portables to parallel supercomputers, all of which can be accessed by users with a common user interface, and which run a wide variety of industry standard applications. The era in which users had to discard all their applications and investment in learning them simply because they purchased a different class of computer will seem as quaint as the distant days when every brand of typewriter had a different keyboard layout.

We have some distance to go before we make our landfall on that friendly shore. And yet the welcome scent of land wafts above the waves and the shorebirds circle above our craft.

What is the true significance of Windows? It empowers.

Windows empowers the tens of millions of owners of industry-standard individual computers with the ease of use, inter-application data transfer, and device independence which have long been available only to those who spent far more money to buy and time to master a high-performance engineering workstation. Windows empowers application software vendors like Autodesk by allowing us, at last, to deliver an intuitive graphical user interface, on-line assistance, and all the other benefits of the workstation environment not, as in the past, to a tiny fraction of our customers, but to all of them—and by doing so to raise their expectations once again and ever higher.

Windows empowers both developer and customer by providing a migration path that protects the investment we both make, a guarantee that the evolution of hardware will not leave us orphaned as has happened so many times in our past. That migration path, evident already in the progress to Windows 3.0, the forthcoming 3.1, and, on the horizon, NT, makes us confident that products such as our forthcoming HyperChem for Windows—a molecular modeling product that would have recently been deemed a supercomputer application—can, evolve, as Windows grows, into the multitasking and parallel architectures that will at last put supercomputer power on every scientist's benchtop.

OK, so Windows is a Big Thing—everybody knows that. But why is computer aided design so important? Because it is the single most important thing you can do with a computer and it will, in time, I believe, become the largest single area of computer application. Some people ask me if the market for CAD is saturated. Are they kidding? Are they crazy? Even if you limit CAD to professional two dimensional, production drafting the market is far from saturation. Every shop with one or two copies of AutoCAD and five or six drafters still on the board represents future revenue for Autodesk. Every company with dozens of drafters and no CAD system is an opportunity to demonstrate the proven productivity benefits of CAD and broaden the market with every sale.

In the larger world of CAD, where the "D" stands for "Design" instead of "Drafting," we have barely scratched the surface. The computer aided design industry is still in its infancy. Even in two-dimensional drafting, most drawings are still done manually on drawing boards. The standards for three dimensional design, solid modeling, conceptual design in architecture, interactive mechanical engineering, facility planning, geographic information systems, and integrated flexible manufacturing have yet to be established. Each is a fledgling market with a potential as great or greater than our current AutoCAD business, and all are poised to grow at an accelerating rate for the foreseeable future. Autodesk intends to become the leader in every one of these areas.

Amidst all the daily news and quarterly Wall Street thinking, it's easy to lose perspective. Only in America can you earn more than 57 million dollars in one year in the middle of a depression and get beat up. Let's look a little

further out for a moment, because what's going on in our industry isn't going to end this quarter, or next year, or in the foreseeable future. The basic trend that drives this industry—the fact that every 18 months the computer power available at constant price doubles—continues intact and may even be accelerating. Look around this room. Other than people and plants, this room and everything in it was designed and manufactured. Today, despite all our progress in computing, the process of architectural and engineering design and manufacturing is done much as it was fifty years ago. All of this is about to change. In the largest companies, you can see glimpses of it already, but, as with the first computers, the benefits have not yet entered the mainstream of the economy.

That is the work before us. Soon, millions of designers around the world will possess the tools that empower them to design the products of the next millennium. In the United States alone, there are more than 600,000 manufacturing organisations. Eighty-five percent employ ten people or less. The revolution in design and manufacturing that is about to occur will change the way we conceive and construct every artifact of humanity. This is not a small market, or one in danger of "saturation." Its size is constrained not by economic forces so much as the availability of computer power and the creativity of the human mind. The first is growing exponentially with no end in sight. The second knows no bounds.

We are living on a small blue sphere in an endless black void. Over the next twenty or thirty years, the human population is expected to double. Five billion new people are going to be sharing this world with us. To provide those people with the food, the shelter, the clothing, and all the other things we feel entitled to ourselves, we are going to have to design and manufacture, over the next several decades, as many artifacts as all of humanity have created over the last two million years. And we shall have to do that in a way that preserves this fragile home of mankind for the generations that will follow.

The mission of computer aided design is to create and deliver the tools we will need to accomplish that. What we do at Autodesk is to build, inside a computer, models of things that exist in the real world. Whether you call it computer aided drafting, or solid modeling, or computational chemistry, or desktop video, or virtual reality, this concept is at the heart of the technological adventure of the second half of the Twentieth Century and will form the centerpiece of the industrial revolution of the Twenty-First.[414] Autodesk's goal is provide the tools for this Golden Age of Engineering to every designer, in every industry, in every nation, so that their creativity can help us all to succeed.

What we are announcing today is a modest yet significant milestone toward that distant goal. It is now my pleasure to introduce John Lynch, co-manager of the AutoCAD Business Unit. John will describe the details of AutoCAD for Windows and our future directions for CAD under Windows.

[414] This passage was baldly recycled from the Neuchâtel Grand Opening speech (see page 709). I figured that even in the unlikely circumstance that anybody in the audience had been there, at least they were hearing it for the first time in English.

MicroTimes

On March 26th, 1992 Mary Eisenhart of MicroTimes *visited Autodesk and conducted the following interview with John Walker. The following is a complete transcript of the interview, which lasted almost two hours and covered a wide variety of subjects relating to Autodesk's history, present, and future. A shorter version of this interview appeared in the May 11th, 1992 issue of* MicroTimes*. Although this is rather long, it's the best distillation of how Autodesk looked ten years into the adventure in the eyes of somebody who'd seen it from the start.*

Q: It would be more standard, at this point in a company's evolution, for somebody in your shoes to have washed their hands of it and walked away. You seem to be doing rather the opposite—while not being chairman, not being president, and categorically denying that you're calling the shots, you nonetheless do not just leave Autodesk to its own devices and let it sink in the mud. At the recent press conference, you were a pretty forceful advocate for the new products and their importance. Why are you doing this, instead of going back to Switzerland and doing something else?

Well, I guess you have to ask the question "Why do people wash their hands of companies?" If you get fired, that's a pretty straightforward case. There's the case where you conclude that "the company wants to go this way, I want to go that way."

Those are all things that I've certainly thought of many times. I'm always constantly reevaluating everything.

Basically, where Autodesk should be going is where I want to go personally. And as long as I can persuade Autodesk to go in that direction, I consider that my efforts are more useful, more effective, leveraged by having the resources of a company like Autodesk going my way, than they would be off by myself in my pump factory in Switzerland.

I'm in Switzerland because that's where our European technical center is. That's why Switzerland. In fact, about this time last year I thought our European technical center was going to be in Nice, in France, and so I expected to be there.

What I intend to spend the next five years of my life doing is working on automated manufacturing. I'm certainly planning to be doing that in my capacity as a programmer at the European technical center of Autodesk. If Autodesk recruits a new CEO, and the CEO's first act is to fire me, I'm going to spend the next five years of my life automating manufacturing anyway, okay?

It's what I want to do, and I believe it's the most important thing I could do for Autodesk. And that whether I do it or not, it is something that Autodesk has to do if Autodesk wishes to remain the strong force in the CAD business that it now is, because that's part of where the CAD business is going.

Coming over here entailed sacrificing a large amount of what I've spent the last five years of my life trying to achieve, in terms of having a low profile, in terms of not being identified with the company. Throwing that away, I've lost a large part of my life, and it's not clear that it's retrievable, at least not without another four or five years. I guess that's an indication of how much I believe in what we're doing here, in what we're attempting to do, and how much I believe I could contribute to achieving that.

Q: You've defined what you think Autodesk should be doing. What other likely directions do you see for it, whether or not they ultimately turn out to be pitfalls?

Well, I think you have to take an appropriately long view of the context we operate in. We are in several businesses, which I don't think you could find any competent thinker who would not believe are going to grow very, very rapidly over the next twenty, thirty, fifty years. If you are in a business like that, then you have to be a competitor who is constantly concerned about how much of that business is yours.

I believe that however you define market share, having the largest market share in the CAD business is an asset that is measured in the hundreds of billions of dollars over the next fifty years. I believe that having a commanding market share in desktop video is an asset that will be worth hundreds of billions of dollars over the next fifty years. Similarly you can just line these things up.

Now if a company focuses too much on the short term, on the margins this quarter, the margins over the next year, and sacrifices market share, that's a bad business decision. That is squandering an asset of the company for a near-term benefit. That's not in the long-term survival interest of the company.

And if you look at companies that are extraordinarily successful over periods of a century, you'll find, I believe, that they achieve that by taking the appropriately long view. The company I would cite who takes that long view is IBM. If you look at IBM, they are willing to lose essentially unlimited amounts of money in technology development efforts that are in their long-term strategic interest. That's what I think you have to do. If you look at the position that Microsoft is in, they don't believe this industry's going to go away next year, and they want to own as much of it as they possibly can, and as broad a base as possible. That's what any competitor has to do.

I think we have a tendency in this business—perhaps it's educated in business schools—to view competition as some kind of a zero sum game. It's not a zero sum game at all. This industry didn't exist in 1976. It all came from nowhere. And a lot of additional things are going to come from nowhere over the next three years, five years, ten years. The products that you're going to be writing about in 2002, probably none of them have been started yet. They're the exciting non-obvious things.

When you have a success, it's easy to assume that 1) it can go on forever, and 2) that that's all you need to do. I look at Autodesk, and I look at our rate of growth, and I look at Microsoft's rate of growth. I look at Borland's rate of growth. I look at the other players in this business, and I prefer not to think of it as a predator/prey relationship, but rather colonies of bacteria in a Petri dish. If one is growing at 75% a year, and the other is growing at 15% a year, then before long you tend to end up with all of #1. Not that #1 even bothers to devour #2, but rather that #2 is sufficiently irrelevant now that it just doesn't matter.

That's what really happened with Autodesk in competing with the major mainframe CAD vendors for the drafting market. We didn't really take seats away from them—most of the companies that had those mainframe systems still have them, in fact have more of them. What happened was, we created an entirely new market for CAD that was the other 98% of the business they weren't selling to, and that's where the growth came from.

I believe that that's what you have to do. I think it's crazy to say that the CAD market as it exists today is all there's going to be in CAD. In the long term, there isn't going to be drafting. In the *really* long term there isn't going to be drafting. But I think getting rid of it is going to take a lot longer than a lot of people believe.

Q: Especially since, as you say someplace, a lot of people are still doing it with pencils and paper.

It isn't even that so much. When we talk about eliminating drafting, we're talking about eliminating a language, a form of communication between the designer and the manufacturer, that goes back to the Pyramids. Certainly the Parthenon—there are places in the Parthenon you can see reference lines cut in the base.

It's a highly effective form of communication. It's a form of communication that is so evolved after thousands of years that it challenges the designer of the CAD system to better it. I can literally get a napkin, and about that fast, make a drawing that I can hand to a machinist who can hand me a part, with tolerances of a thousandth of an inch, to my specifications. Block, circle, just draw the dimensions, it doesn't have to be precise—that person will manufacture it for me. If I were to sit down today using the best solid modeling systems in the world, it would take me a comparable amount of time to build a model of that object as it takes the machinist to make it. Now if I'm the designer, which am I going to use? If I can scribble down a couple of lines and give it to some guy, and he goes off and does it, why should I manufacture the part myself so that now it can be manufactured?

That's the challenge we have. It's really as if we had word processors that worked with proofreader's marks still. As if we had all this computational capability, but basically you printed out a paper document, marked it up with proofreader's marks, fed it back into the scanner, and then an hour later you got another draft out, or something.

We've changed the way we write because we have word processors. We haven't changed the way we manufacture because we have CAD yet. That's not going to happen in five years. It's going to start to happen in five years. In isolated places you can see it now. In the very high-end military systems they're building fully faithful 3D models of the objects before they're ever assembled, they're doing automated manufacturing takeoff from these models, and on a project the scale of the B-2, they save money doing it. But for that technology to filter down to the machine shops, most of which are very small and don't have a large capital budget, is going to take a long time. And it's going to take pushing.

Those were the very people who, twenty years ago, were putting in CAD systems for drafting. So that's the filtering-down process, and I don't mean filtering down passively, I mean people like Autodesk pushing that technology out into the world, and creating opportunity for these people. Because it's fundamentally an issue of productivity gain. It's productivity gain, and that's survival in the manufacturing business.

Q: Is this one of the areas where the Cyberspace project would tie in?

It's central to why I encouraged Autodesk to start the Cyberspace project.

I recognize that all kinds of people say different things about this. To me, Cyberspace isn't William Gibson's vision, it isn't the global net. When I use the word, I'm simply talking about a 3D user interface. That is all I'm talking about when I use the word.

Q: I haven't seen it for about three years.

It hasn't changed much, in what you see in the demo. What underlies it is a lot more real now than it was then-real, and portable across hardware, and so forth.

It's a user interface technology. One of the things I wanted us to do two years ago, and maybe when we get the product finished we'll actually be able to do, is to have the world's first cyberspace race.

What I want to do is to take a simple mechanical part, like a universal joint for example—two Us and a couple of pins—and set up three workstations, as they were, side by side. One has a very conventional 3D CAD system with a tablet interface and a flat screen. The other has a nice shaded 3D fast rendering engine on a flat screen, and a dial box and various things you use in high-end mechanical CAD. And the third has the Cyberspace station, whatever technology we're using. The guy fires the starting gun, and three people, from all the parts of this thing, assemble it in space, which is basically positioning operations. Position this so it aligns with this, put this piece in here. This is something you can do on any CAD system in the world.

If we've got Cyberspace right, the guy with the Cyberspace rig is going to be able to do it in about ten seconds. It doesn't take long to put a U-joint together—you just pick it up and [whoosh] you're done. With a great graphical user interface—click, drag, position, pull down menu, dialog box, drag, align—it's ten times longer. And if you're working on a tablet with a 2D, you're typing in all these numbers and angles and aligning this and verifying that, and looking at it from a different angle. . . .

That seemed to me to be the big win in Cyberspace. To me, the relationship between Cyberspace and CAD was the same as between drafting CAD and high-performance graphics. In the early days of the graphics business, back when we had Adage displays and huge Tektronix storage tubes and other things that cost large amounts of money, they were bought by the very first high-end CAD users. Because as expensive as graphics were in those days, there was a clear productivity gain in doing CAD with graphics over doing CAD without graphics. CAD was done without graphics. There were people that did computer aided design with FORTRAN programs and plotters. There was a whole generation of CAD that was done that way, that we kind of forget about. The graphics screen was an obvious huge productivity gain.

Similarly, I think the early adopters of Cyberspace are going to be people doing 3D modeling and 3D positioning. And ranging all over the design spectrum, certainly not just mechanical; architectural walkthroughs are an obvious application. Really, across all of 3D design. That's an application that can really bootstrap this hardware to the point that it becomes accessible for lots of other people.

I guess time has proven me wrong, by the way. That's why I thought [virtual reality] was going to happen in 1988, and a lot of other people were saying "No, it's entertainment that's going to push it out first." And entertainment has. There's much more VR hardware both being developed and being used in entertainment applications than in design at this point. The design applications tend to be fairly arcane, and certainly very rare. You see some people doing some things like maintenance accessibility and so forth.

I guess one could also say that the VR hardware we have is so bad that a designer can't really use it now, and that's part of the problem.

It's like, you can play a game with a Nintendo, but you need better resolution than that if you're going to do serious design. And I guess we're kind of at the Nintendo level.

Q: If you're manufacturing sensitive parts, Nintendo is not what you need.

Right. And it's also a question of how complicated things can be. Part of what people miss about CAD versus other applications is—a 12-, 15-, 35-megabyte CAD database is large, but not huge. Designs are very, very complicated things. And the CAD user, in fact, always wants to have more and more speed and complexity, simply because they don't want to have 50 separate models that are subassemblies, they'd like to have it all in one model. They'd like to have one faithful model.

So the CAD user is always trying to get more CPU power, more graphics resolution, more speed, simply because of the productivity win of being able to get larger models in there. When you start talking about VR hardware that can't track in real time when you have 2,000 polygons—2,000 polygons is *nothing* in the CAD business. We're talking millions. Tens of millions.

Q: What is Alvy Ray Smith's definition of reality being a certain number of polygons?

Eighty million comes to mind, but I'm not sure. I think that's based on the human visual field, and basically if you assume one polygon per pixel of your human resolution, you end up with something like that. The intriguing thing is how close we are to that.

But of course, Alvy's reality is 80 million polygons at sixty frames a second or faster. We're a *lot* further away from that. (laughs)

Q: Where does multimedia segue into all this? Where do sound and motion become important?

Well, I never thought of sound and motion as media. I don't use the M-word. I prefer to describe what we do in our division that bears the M-word as desktop video.

I think desktop video is a huge business—again, it's one of those businesses that nobody is going to take seriously until the hardware appears that really does the job. And when the hardware appears, it is going to be *so obvious* that it's going to change the world. Vast amounts of money are going to be made, people are going to do things in different ways, people are going to learn new skills.

Today we're really at the Altair level of desktop video, just about. The hardware is something that an enthusiast can get very excited about, and with a vast amount of suffering can do amazingly impressive things—if you're another enthusiast.

We all remember back in 1976—"Wow! Look! It plays Tic-Tac-Toe!" "Wow! It computes the first eight digits of pi in six hours! That's really cool, isn't it?" "Yeah, yeah, that's cool!" (laughs) It took a while, and everybody who "got it," to use Ted Nelson's phrase, understood that in five years these things were going to completely change the world. It would change the way everything was written—when you have a computer, you don't write the way that you write when you have a [dedicated] word processor. They understood that someday, modems would connect all these things together, and there would be a global community.

But that [laughs] took a lot of imagination. You had to be plugged in to what had been done on the high end, to see the potential, and you had to believe in something that I think for a lot of people is very difficult to believe in.

There aren't many times in human history that you can draw a straight line on semi-log paper over forty years, and have that trend continue. If you talk to people in a normal business and say, "Well, okay, yeah, a car today goes 55 miles an hour, but in ten years it's going to go half the speed of light," they kind of say, "What are you *talking* about?" But here we say, "Well, yeah, sure, we design something [assuming] everybody's going to have 32 megabytes in their PC in three years." I have 32 megabytes in my PC today, and it cost me less than my Altair.

It's really mind-boggling. And if you really accept that, you can buy into technologies that today don't look terribly promising at first glance.

Desktop video—how big is the market for this? Well, how many camcorders are there out there? You own a camcorder for what, an hour, before you realize, "Gee, it'd be really nice to be able to edit this stuff." "How

do I do that?" Well, you can't. "What do you mean I can't?" "Well, you can buy all this special hardware and it still doesn't work." "Why can't I just do that on my PC?"

Well, why can't you, after all? You can go to Ted's houseboat and use his Aavid system. Clearly everybody's going to be able to do that on their PC. The Aavid's s a Macintosh, after all, with some special hardware in it. That technology is going to be everywhere.

It's a new art medium. We're talking about the ability to do Disney, Lucasfilm—choose what you like—3D animations, 2D animations, multi-plane animations. You don't just make a new art medium available to a lot of people and suddenly expect a million people to run out and start using it. I'm sure it took things like frescoes a while to catch on also—the technology has to get diffused. . . .

Q: *"Wet plaster? Are you nuts?"*

[laughs] Egg gesso. Talk about low tech. . . . When you make a technology like that available, you can't expect suddenly to have everybody start using it. You sell the technology, and every trained animator in the world buys it. Okay, well that's a hundred and fifty. What about the next million? Well, you've got to get it into the schools, you've got to get books written about it, you've got to show people what they can do with it. You've got to give them something to start with so they aren't faced with a blank sheet of paper. After all, you tend to start with a coloring book rather than a notepad, and that's a lot of what we have to do to get it primed.

But in the long term, how can it *not* be something huge?

But I don't think there's going to be a multimedia market any more than there's going to be a graphics market. In the '60s and early '70s, there was the computer graphics industry—the National Computer Graphics Association, the SIGGRAPH show. Analysts would tell you how much market share each of the players had in the computer graphics industry—Adage and Sanders and all these people were out there, Tektronix owned a huge piece, CalComp owned a huge piece with their plotters. And that was the computer graphics industry.

There isn't any computer graphics industry any more. It's gone. It's in the box. Plug in the PC, and there's graphics. If it doesn't have graphics, it will in a year. So there's no graphics industry. Those people that sold that stuff in the graphics industry, for the most part, are still around, selling what's now the high-end stuff. But that's not the graphics industry. Graphics is part of the PC.

Pick a number—how many years is it going to be? Every PC is going be able to read and write sound and read and write video. It's not going to be as useful if it doesn't have that capability. And I think we're really there. Every Macintosh has had sound since Day One, and I think really every Windows machine is going to have sound very soon, except in large corporate applications or something like that. It's just a matter of time before the chip's on the motherboard.

We spent the first three years around here trying to convince people in this grand and glorious world that you needed a floating point coprocessor. We used to have a version of AutoCAD that ran without the floating point coprocessor, for longer than I either remember or would like to tell you. You plug in the coprocessor, and the product runs five times faster. Not until the 486 was floating point a standard part of the microprocessor—and what's the first thing they did? Take it out! Make a 486SX that doesn't have floating point, and then sell you this floating point that's the whole chip! It's crazy!

It's amazing to me that you can do anything at all without floating point. In the CAD business, basically every pixel you see on the screen has gone through a 4X4 matrix multiply. And how patient are you? Particularly when the cost is so low.

So we saw that with floating point. We've seen it with graphics being in the box. From the very early days, we sold very well on machines that came with graphics. For the first year of Autodesk, Victor 9000 outsold IBM, because the Victor 9000 came with 800×400 graphics right out of the box. And the IBM—hey, remember the CGA? First you had to buy one, then you had to get it to work, then you had to put up with four colors—

So really, once something is in the box, the software will appear. People will start to use it. And I think we're really getting to that point on the hardware you need for desktop video now.

But again, we're clawing our way up a very tall ladder where the rungs seem to be very well greased. We're going to get sound hardware on the motherboard, I expect, within the next generation of motherboards. We're what, maybe two or three years, from having the JPEG chip as a standard component. All these little pieces have to come together. But they clearly will.

I never quote numbers in these things because I'm worse than Reagan on numbers—quotable numbers, I mean—but one of the things we actually did market research on was what percentage of people who own a camcorder own a PC. And it's a huge percentage. If you told people that you could do useful editing, useful production titling, with your PC, a large number of people would, and that's a number in the tens of millions. That's the size of this thing.

There's this feeling that all the money in multimedia is going to be made in titles, in CD-ROMs. I'm not sure that's true. I think we may be trying to apply the TV model of the passive receiver of information, to something that's a creative medium. I'm sure it'll exist, but I'm a lot more interested in getting the creative tools out into as many hands as possible, because I think that's going to cause this upwelling of software.

Again, we've seen it before. How can you talk seriously about selling tens of millions of Pascal compilers and C compilers? Nobody wants to be a programmer! And yet people have bought tens of millions of Pascal and C compilers, and it's changed the world. Because anybody can program now. You don't have to work for a big company and do what they say. And look at all the software that's been created by these people.

That's what we need to do with these video communication tools. And we're just at the beginning of a terribly long cycle here.

Q: Do you see that dovetailing with CAD at any point?

Again, I think the CAD user will be the early adopter of many of these technologies. We're seeing it already. Clearly, architectural walkthroughs are a major application of 3D Studio. Product illustrations, product concepts, are places that CAD and multimedia/desktop video fit together very well.

A substantial number of our installed base are people who either already have AutoCAD or use AutoCAD within an organization, and I think that's a toehold that can make us successful in this market—that we can then broaden out, I trust, to the larger market. I think the parallel with Cyberspace is very similar here.

To get very far with this stuff, you've got to understand 3D modeling. And there aren't a lot of people in the world who understand 3D modeling, except CAD users. And so you don't have to bootstrap them up through the whole concept of building models, thinking in 3D, working in 3D, handling lights, handling rendering. They already know that. They're doing it. So to a CAD user, you're simply saying "All you're really doing is adding motion and dynamics" to something they already understand. If you're talking to somebody who watches MTV and would like to do stuff like that at home, you have a much longer learning curve.

Clearly the people that first bought PCs were people that already used computers at work. It took a while to broaden out. But it's inevitable that it will.

Q: You've said something I thought was intriguing—that there's a perception that CAD is a vertical nichey market, but that after all, there was a time ten years ago when spreadsheets were considered a vertical nichey market. How do you see the analogous transformation taking place?

Well, CAD is about building models of real-world objects inside the computer. That is what CAD is. I believe that in the fullness of time, every object in the world, manufactured or not, will be modeled inside a computer.

That is a very, very big market. This is *everything*.

Look at the huge successes in digital technology. It's basically been making things that used to be analog digital, right? In some finite number of generations we won't have to suffer through this [photo shoot] because there will be digital cameras that are good enough to do that. Sound went digital in the 1980s; it was one of the huge consumer electronics successes. Musical instruments went digital, with MIDI. I think we see on the horizon that television is going digital, with whatever technology they adopt for HDTV.

And every time you go digital, whatever the original motivation may be, you end up with a tremendous number of ancillary benefits. The fact that the data are in a form that one can process them fundamentally changes things.

What CAD is about is that artifacts are going digital. In Arthur C. Clarke's book *Profiles of the Future*, in, I think, 1963, [415] he had this timetable of the future for the next 75 years. One of the things he talked about, I believe in the early 21st century, was encoding of artifacts. In other words, storing artifacts in digital form, and potentially replicatable form. That's what CAD is about. Today we're in very, very narrow areas of the whole field of being able to design and manufacture anything using your computer.

Again, it all happened before. We had a word processing business before we had personal word processing, and it was in a very vertical market. Large corporation typing pools, legal offices, insurance company client writeup—they had a word processing industry. All the analysts analyzed all the pie charts of market sector, and they missed the fact that there was a market for about another half billion word processors that weren't in the word processing market.

That's what's beginning to happen with CAD. We've seen it over the last ten years, but we haven't really seen anything like the takeoff yet. If you have lots of money and are very patient, you can get a stereo lithography system, set it up on your desk, build a 3D model of something, push a button, and after an embarrassingly long time, get a piece of plastic that is a 3D, faithful-within-the-resolution-constraints-of-this-device technology. That's desktop manufacturing, and again, that's worse than the Altair level right now in terms of cost and time and plastic that you use and so forth.

But I expect in the ultimate destiny of this that there's going to be an appliance that sits on your desktop that can make anything. I don't expect that to happen within the lifetime of AutoCAD Release 13 or 14. But that's where it's all going.

This week we're shipping HyperChem, which lets you design objects at the molecular level. The technology is going to go on scaling down until we have the same level of fine-grained control over the structure of matter that biology has been using for the last billion years or so. And at that point, we're going to be able to make *anything* from a digital model.

That's going to *really* change the world. It's going to change the world beyond comprehension, I think. Beyond, perhaps, the comprehension and imagination of the people who are spending a lot of time thinking about that

[415] *Profiles of the Future* by Arthur C. Clarke, Harper & Row 1963 / Bantam 1964, 1967.

technology.

And central to all of that happening is computer aided design. You can't do anything without building a model of it. If you're building a workbench, even if you're doing it in your head, you have a model of it in your head. Something tells you how long to saw off the plank, something tells you how many supports you need. You have a model of it.

Well, doing models in a computer is going to make the whole process much simpler, eventually. When we figure out how to communicate these things, when we have the right user interface technology, when we have enough compute power to throw at these problems.

As the microcomputer has evolved, we've been picking off the easy things. Word processing, the personal database, the spreadsheet. I think the thing that kind of holds what Autodesk does together is we're doing stuff that's going to use up all those compute cycles over the next 20 or 30 years.

If you go on drawing that line on semi-log paper, and you assume that in 1998 you have the power of a Cray 2 on your desk—what are you going to use it for? You're going to use it to do things that are fundamentally different from what you use your Macintosh for, and are the kind of things that very, very few people are doing with Cray 2s right now.

That's what we try to identify—bet on the technology going on, start to build these ambitious things that can take advantage of the computing power as it arrives. That's what we've done constantly in CAD. I think we see it in desktop video. We certainly see it in molecular modeling. Xanadu's great today, but it's about 100 times too slow. Well, time will solve that. And those are the kinds of technologies, to me, that are a lot more interesting than going out and competing for market share points in the spreadsheet market or the word processing market or the database market.

These are things that don't exist at all today. It's hard to keep focused sometimes, as the years pass, on getting these things finished, and to keep the belief that it's going to happen.

But we have an existence proof. AutoCAD was not viable at the time we started working on it. There simply wasn't enough compute power around to do the job. The graphics were laughable.

Time solved that. And if time hadn't solved that, we wouldn't be around. So those are the kinds of things that I prefer to look at

As a strategy it can't fail. The strategy can only fail if you don't believe in it or you don't execute it properly. Do we really think that this evolution of technology is going to stop in the next two or three years? That suddenly machines aren't going to get any faster or have more memory?

If that happens, you'll be able to see it coming because you'll be able to look at the fundamental technology and say, "We can't go any further without a fundamental breakthrough." But nobody sees anything like that on the horizon for at least 20 years. There's a point where you have to look at quantum electronics and so forth. But even just the linear progression of the stuff we've got is going to carry us into the 21st century.

To me, plotting the strategy for a software company isn't very difficult. Particularly once you have a success like AutoCAD, because you have an installed base that is beating you over the head—"We want this, we want that, we want this other thing. We want you to do this. We also want this." So you don't have to do an enormous amount of market research when you've got 600,000 people out there asking you for things that you don't have. It's basically a question of deciding what to do first.

When you have the financial strength that we have, if it costs you a quarter million dollars to do a product, and one in a hundred succeeds the way AutoCAD has succeeded, we can do a hundred failures for each success. The ratio of what it costs you to build and launch a product, even today, to what you make when you hit, is the thing that makes this business totally unique.

This isn't like the chip business. It isn't like the steel business. It isn't like the auto business. It's like the movie business. You make lots and lots of movies in the hope that, if it's a great year, you'll have one blockbuster, and if it's an unbelievable year you'll have two blockbusters. But you probably made 18 movies that year. And you may go two or three years making 18, none of which is a blockbuster—they may be profitable.

I think that's the kind of business we have here. It's a business that evolves rapidly, where preferences are difficult to predict, and yet failing to innovate is not the way you make a blockbuster. Making the 900th Western doesn't give you the kind of success that making *Star Wars* did.

That's what I always look for. I think that every company that tries that—unless something exogenous happens to blow them up or they lose their vision—they all seem to succeed.

Q: How does a company keep its edge, or its soul, or whatever you want to call it, in the face of overwhelming success?

I haven't solved that problem yet!

Q: I know. That's why I'm asking you, because you're one of the few people I know who seem to think about it at a meaningful level.

Well, no, I think if you look at companies who are successful over a long period of time, you find that there are people there who think about those things. I'm very sure you find it at Microsoft. I'm very sure you find it at IBM.

It's a terribly difficult challenge, because when you build a company, you have to be faced with the reality that as you grow rapidly, at any point in time most of the people you have working for the company are people who arrived after the initial success that created the company.

What is really necessary is to provide the message and the history of the company, so people understand it's not like it was always there. There was a time at which it wasn't there, and there was a time in which it appeared rapidly. That's why I did *The Autodesk File*. It was not really intended to be circulated outside the company—I seem to have that problem a lot [laughs]. It was really just a collection of stuff that people who were interested in where the company came from could use to find out, with the goal of leveraging that experience to people that arrived too late to have lived through it.

The other part, though, is really self-fulfilling, in the sense that if you go on having successes with new products, then you have cadres of people who have seen it happen. And so it is essential that you do what is necessary to make successes of your new products. That's what I'm trying to do right now—guarantee that every product we invest in will turn into a success, if it has the fundamentals to be a success. Again, I'm willing to fail. I'm *not* willing to invest in a product and then never give it a chance—pushing it off the loading dock, as I refer to it.

So this product [HyperChem] is going to be one of the biggest rollouts we've ever done. The AutoCAD Windows rollout[416] obviously got us a lot of attention. And all of the other 29 products that we're rolling out

[416] See page 781.

this year will be rolled out in a manner distinctly different than the Autodesk tradition, and I believe that more than one of them will be a success.

That changes the mindset of people. You get into this depressed "everything's going to fail" mode. Everything doesn't fail. Companies have lots of successes. But you have to believe in it, and you have to make it happen.

It's not like the product comes out on the end of a conveyor belt and God looks at it and if a lightning bolt comes down you get all this money. [laughs] You take it off the end of the conveyor belt, you go out in the world, you start jumping up and down and waving it in front of people, and finally convince them that it's worth giving you their money to make you go away [laughs]. And then they take it home, and "Gee, that's really cool!"

But you have to keep pushing. At Autodesk I think there was a mindset that the machine runs all by itself. That it's just magic. If you look at the companies that advertise and promote most heavily, they're the people that make this [can of Diet Coke]. They're Kellogg's. You're talking about products—in the case of Coke Classic, the product has not changed in over 100 years, and the one time they changed it they regretted it. Kellogg's Corn Flakes—how often do they re-engineer that product? Release 3.7? No, no, no, sorry, it's Kellogg's Corn Flakes. And yet they spend billions of dollars promoting those products, because they believe that in the year 2700, if there are still human beings, they will still be eating corn flakes, and they will still be drinking Coke. And they want to own that market, or as large a part as they can.

It's not a self-feeding process. You have to explain the benefits of your product to people. And you have to do it on an ongoing basis as you improve the product.

Q: A lot of it seems to be mutually exclusive loops—you've got the people developing the product, you've got the people marketing the product who are completely clueless about the people who are developing the product, and so it goes all down the line.

Well, you have to, at some level, just have some basic rules, Okay?

I feel kind of embarrassed because I often use the phrase "you don't have to be a rocket scientist to figure this out"—and here I've got a product [pointing to HyperChem box] you really *do* have to be a rocket scientist to use [laughs].

But I never used to understand, back when I was a big-computers kind of guy, the concept of the salesman who didn't understand the product. The salesman who couldn't demo the product. The programmer who didn't know the names of the competitors of the company, who wasn't familiar with the competitor's product.

To build a company that's a competitor, you have to have everybody be a little microcosm of the company. You've got to have the programmers be intimately aware of what causes the purchaser to choose their product or a competitor's product.

Now that may be marketing, or that may be sales, but if you're a developer and you don't know that, how do you know what to develop? How do you know what your target is, what your priorities are? It's not looking at the competition to be your guide, it's knowing what's in the customer's head. Knowing who the customer is. Knowing how the customer obtains the product.

Similarly, if you're out selling the product, the people you're talking to who are buying the product know about your product, they know about the competitor's product, they know about where the market's going, and if they ask you questions about things like that, and all you know is how to fill out the sales form, how are you going to sell?

So I think that you really need to imbue in the whole organization an in-depth knowledge of what this company does. I think it's got to be across the board. That is part of the reason that Information Letter 14[417] was so long, and that I wanted so many people to read it. It was because everybody who works here, here in Sausalito and around the world, needs to understand this business we're in, the industry we're in, and what's happening to it over the next ten years.

Information Letter 14 was not intended to be a secret—no more than the first twelve were. It got out in an earlier version—in fact the one that most people have read is a draft, not the final one, because that's the one that started self-replicating. Some things changed—there were some things that I thought weren't right in the first draft.

But it was inevitable that it would get out, and I intended it to get out. There wasn't anything that I said in there that I had not heard from analysts who covered the company. There weren't any secrets in it. It was really just a view of what the company had to do, that I think you cannot communicate to five people and change the outlook. You have to communicate it to everybody who works in the company, or else you have no chance.

But I certainly learned a lot about publishing. I mean, I sweated blood over this thing [*The Autodesk File*] to get this book put together. Hey, I just leave something on somebody's chair, and before you know it it's all over the world! I guess I'll do that with my next book, *The Hacker's Diet: How to lose weight and hair through stress and poor nutrition*.

Q: Where does AMIX fit in?

AMIX is, again, I believe, one of these little hard-to-define, hard-to-explain things that, once people experience it, is going to be so obvious that it is going to grow very, very rapidly. I really believe, of all the things we've done, AMIX has as much potential to change the world as anything we've done. And even having watched it be built, heard the rationales, watched it be put together and grow—experiencing it is fundamentally different than what you expected. It's like you can imagine what it's like to be on the WELL, but it's very different to *be* on the WELL.

For years I tried to imagine what it would be like to be able to read netnews. We were never on netnews, and then finally we were—and it was nothing like I expected. And that's really what AMIX is.

I've sold stuff on AMIX—actually I have a net profit in my account now—and it's very intriguing to realize, "You know, I do things very differently, even if I'm going to ask two dollars for it, than I do when I'm just posting something to a newsgroup." I put an entirely different amount of work into it, I'm much more concerned about my reputation and the comments that are going to be put on this thing. I make tremendous effort to target the introduction to be both persuasive and fair, so nobody will buy it that isn't going to be satisfied with it.

The anonymity of AMIX is very interesting. Like being in a bookstore and knowing that you've sold 22 copies of something, but not know who bought them. They can communicate with you, but the fact that you bought something does not generate a mailing list in the same sense as a bookstore.

And similarly, I think, when you're on it as a buyer, you recognize that the sellers are thinking that way. The sellers that do not think that way are going to be selected out very rapidly, because if somebody's asking you for $15, a book-type price, you're going to read the comments file. You're going to look at the resumes of the people who put those comments there.

[417] See page 600.

Again, the "vision thing" is being able to imagine what something is going to be like before it exists. I've heard that from Ted over and over again, that that's a very rare talent. I really think that Phil Salin knew what AMIX was going to feel like, at least after he'd built the prototype. I think only a very few people really knew what it was going to feel like.

And then once it's built, after that vision, now I think people will evaluate for themselves the benefits of it.

But I think part of it is just fundamental economics. Phil was an economist. He understood economics. He understood transaction costs. He understood the way purchase decisions are made. The trend in price in the software business has been down, down, down, down, down. And we have a means that you can sell software for $2, $5, $10, at a lower transaction cost and a lower distribution cost. The rules limit this, but you could sell software for $50,000, three copies a year, and have a very low transaction cost and a very low support cost. Companies that sell software for that kind of price tend to have large organizations that could largely be eliminated if they could deliver their updates and support and consulting electronically.

These are all things that AMIX is going to be getting into. We're getting a lot of interest in AMIX purely as a fulfillment mechanism, of being able to sell an update. Your update to Product X, Release 4.0, for $10.

Part of the problem we have to climb up against—and I think Prodigy actually helped a lot in this—is that surprisingly few people have modems. We looked at some of the demographics, looking at AMIX, and it astounded me. It was a shockingly small number.

Slowly people will realize the benefit, but they're not going to buy a modem until there's something that they want to talk to.

Q: And by the same token, a lot of people won't buy Macs until the Photo CD is out and the Mac reads them. All these years I've been telling my parents they don't really need one, because they're not into the toy aspects of it at all. But as soon as they can put three generations' worth of family pictures on a CD, then everything changes.

Mm-hm.

Q: Are the AMIX guys also Xanadu guys? I always see them in the same place, so I guess I've got that perception.

No. No. They share a parking lot, of course, and they share a lot of the same intellectual groundings and ideology.[418] But there is no connection between the two companies. They are two independent companies in which Autodesk has unrelated investments.

A lot of the connection was that when Phil started to think about AMIX, he became very interested in Xanadu, and saw Xanadu as the ultimate system that would underlie AMIX. It would really be the first system that would have the strength to do all the referencing and linking that he would like to do.

In the early days of Autodesk's involvement with Xanadu, he was a very strong force in causing the deal to come through, getting Xanadu organized. He probably spent more time on Xanadu the first year than AMIX. But we concluded that AMIX did not need Xanadu beneath it. An AMIX with Xanadu beneath it will be something very, very different and much better than what we have today, but what we have now could be built

[418] For a taste of these "intellectual groundings and ideology" check out "Great Mambo Chicken and the Transhuman Condition" by Ed Regis. Reading Massachusetts: Addison-Wesley, 1990. ISBN 0-201-56751-2. Freeze your brain, download to nano-bot, live forever, explore the ends of space, know all, party 'till the end of time.

as a conventional program, and it needed to be built as soon as possible without waiting for Xanadu in order to succeed.

Q: I take it you think Windows is pretty important.

I think it is of *surpassing* importance.

What is happening is that workstation-class operating systems are becoming available for the masses. Workstation-class operating systems I really define as a system in which graphics, and soon 3D graphics, are tightly integrated within the system. It has a modern user interface. It has true multitasking, and it networks well. That's what you really get today when you buy a workstation. You have to suffer a tremendous amount when you buy a PC, to stack pieces on top to get all of that stuff to hang together.

When that solution becomes available, the software environment is going to change in a very substantial way. You can certainly say, "Well, what about the Macintosh?" Well, before System 7 the Macintosh was not a workstation-class system. Only with System 7, which I guess a lot of people still aren't using, did the Macintosh become a workstation-class platform. And with Windows NT, we will see the PC become a workstation-class platform.

Suddenly all this capability that people have not been able to afford is going to be available. And it is going to change the way everybody builds their applications, buys their applications, and uses their applications.

I don't know for sure, and at some level I don't care, who wins—whether it's Windows NT, or, should Microsoft stumble, an OS/2 32-bit version. Suppose both Microsoft and IBM stumble, and Sun Solaris for the 486 ends up filling that niche. Or all three of them stumble, and OSF/Motif or something like that becomes a standard. Whoever wins, it's going to happen.

It's as inevitable as anything that I've ever seen in this industry that's going to happen. The benefits are simply too huge, and the costs are going to be negative, really, in the sense that when you go into the stores a couple years from now, there isn't going to be much raw DOS software for sale. It's all going to run under the winner of this operating environment battle.

Any software company that doesn't really understand the true significance of this is not positioned to be a survivor across this transition. You have to anticipate both what's going to happen—make some bet on who's going to win, certainly, in terms of where you spend your time—but most of all recognize what the expectation of the user is going to be. Because the user who has a multitasking, multiprocessor, high-resolution system that's networked to lots of other people is going to be using your application in a very different way than the guy that's looking at a black bozo screen one application at a time.

That means that you have to start making fundamental changes in how you build your product, in what you put in it, in what capability it provides to the user, that are bets on that occurring. Because if you don't start now you're going to be too late.

Q: So you think that, nice interface aside, the main advantage of Windows is that there's so much of it out there? Or does it have other particular strengths as an operating environment?

Well, I think that one cannot ignore the impact of something that has sold nine million copies. Market share counts for a lot, and it counts for a lot when the incremental cost is virtually nothing. If you have a PC with appropriate hardware, it costs you almost nothing to get into it. You're not making a purchase decision like going to a Mac or going to a NeXT or something like that. You almost can't afford not to.

After all, if you want a really nice word processor today on your PC, you've got to put Windows on it. Because all the word processors for Windows are a lot better than any one that doesn't run under Windows—that's my experience. All the nice spreadsheets run under Windows. So you have to do it.

Now when that standard is created, we're going to have software that can move upward from all the software that's being built for that environment to systems of workstation class and above. NT is going to run on symmetrical multiprocessors. Sun is just getting out a system that runs on symmetrical multiprocessors.

I don't make predictions about hardware, because I'm always wrong, and I don't know very much about hardware, even though I used to have a hardware company. But it's not inconceivable to me that in 36 months a machine with fewer than four processors is considered as passé as a machine with no more than 640K of memory on it. After all, the chips don't cost very much, and they're plummeting in price. And if you have software that runs six or eight times faster if you put in six or eight processors, you're probably going to put in six or eight processors, if you're interested in going fast.

This software [HyperChem] will run eight times faster on an eight-processor machine than it runs on a one-processor machine. The SGI version demonstrates that. We're going to be shipping that one later this year.

Dr. Neil Ostlund, who designed this software—he is a professor both of chemistry and computer science; he learned computer science in order to parallelize computational chemistry. You don't just sit down and do this stuff; it isn't easy. It's like you've been writing COBOL business applications for 45 years and then "here's a Macintosh, design a GUI." Well, it requires a little head-scratching and a little learning.

But application vendors who wish to play in this environment are going to have to learn to do it. Particularly if, as in the CAD business, you can use any amount of power. We could saturate a machine that was a thousand times faster than anything you can get today, just doing the same kind of stuff we're doing now. So we'd better be able to take advantage of it.

Don't forget the multiprocessor/multitasking aspects of this. I think it's really a very important thing. And all the major players are going to provide it. It's going to be essential.

Q: In niche markets, or everywhere? For what I use my Mac for, I don't see the need for multiprocessing, but will that change?

Well, are you happy with how fast it runs now?

Q: Depends on what I'm doing. For text, who cares? But for PageMaker, I go crazy because it's so slow.

Okay, well, if you go crazy about how slow it is, and the Mac toolbox were running on one 68040 and the application were running on the other, it would immediately run twice as fast, assuming that the breakdown were about half and half.

I just got my 486 this week. I've been using the same 386 20 MHz machine for, I guess, four or five years. And for everything I do with the machine except CAD and HyperChem, I was quite happy with how fast it ran. Yeah, it took about seven hours to compile all of AutoCAD, but I only do that every couple of weeks. Besides, it can do that overnight.

But except when I was really beating Excel to death or something like that, or editing a 175-page document, I was fine. But as soon as I start to do CAD—"Oh, this thing is so *achingly slow*!" All the floating point-intense stuff, huge databases, lots of operations where you're searching through a lot of stuff to try to find things. . . .

Q: And not even doing 3D motion yet...

That's right! It's basically 2D CAD for the most part.

Our company sold a million copies of AutoCAD and lower-cost CAD. That's not a niche to me. If it is, I hope there's a lot more! [laughs]

I think, yeah, people are going to be doing things that are fundamentally different as well. But they're not things that you can't think of. It's simply things like photo composition. Photo composition and retouch from your Photo CD. You can use a lot of compute power when you get into the really serious image composition type stuff. Video throws another factor of 1000 at it. When you have a machine that lets you cut and paste moving video, you're going to probably use all the processing you can get, and all the processors you can get, to do that.

The other thing that's happening with these workstation-class environments is that for the first time, applications can live beyond the hardware. I have now had pieces of C code that have run on three different processor architectures from Sun, to which I did nothing other than simply recompile. I started on a Sun 3, 68020. I moved to a Sun 386i, 386 architecture. And now I have a SPARCstation. And I did nothing. I just use the code. I don't even think about it. It's inconceivable to me that as long as there is a Sun, I won't basically be able to take my code and move it along.

Lacking that is something that always seemed to me so foreign about the PC business. Because I grew up as a UNIVAC mainframe programmer. I worked from 1967 through 1978, and actually I did consulting into the '80s, on machines that ran programs that were compiled in 1963. Think about that. Basically children running code that was compiled by their fathers. IBM mainframes have been upward compatible through the years—you have to look at the DOS to OS bump in the '60s as an aberration. Yet here it seems like all your investment is a wasting asset that disappears after 24 or 36 months.

You talk about your parents putting all their pictures in the Photo CD—I hope they keep the pictures, because in the year 2700 you're still going to be able to look at the pictures, but what's going to be around that reads the Photo CD?

It makes me sick to think of all these people putting their 8mm movies onto VHS tape and throwing the movies out. And what is there that's going to accept a VHS tape in ten years, or fifteen years?

If you look at the heavy CAD systems, the CADAM, CATIA, Computervision, and the rest of the design software that isn't what you'd call CAD—the structural analysis, NASTRAN, ANSYS—these are packages that are approaching 25 years of age, that have massive investment every year over 25 years, of doing more and more and more. Larger models, better user interfaces. When you make a commitment to something like that, that's a long-term commitment. You and the vendor are taking a very profound step.

Because in engineering, we talk about things that span decades. The Boeing 777 is going to be flying in the year 2050. It'll probably fly till 2075 in the secondary and third-tier market. And they're going to be manufacturing replacement parts for it; they're going to be upgrading it; they're going to be stretching it; they're going to be replacing the avionics and all of this stuff, but that's a design that spans a human lifetime from conception to obsolescence. And you don't want to buy into a design system that's going to disappear somewhere during that cycle.

So there is a somewhat different perspective. This is engineering that we do here. Engineering is unlike a lot of other businesses because when you get a wrong answer people get killed. And when you're educated to be

an engineer, you take a very different view of quality and accuracy and so forth, that's kind of indoctrinated in one.

If your spreadsheet gets a wrong number, you may be embarrassed. If your word processor crashes, you're going to be furious. But if a CAD system produces the wrong strength number for a load-bearing member in an aircraft wing, airplanes start falling out of the sky. And that's really, really bad.

Sometimes we test to death. We're accused of really bashing on the quality of the product. But it can't get wrong answers. It's a fundamental thing. You can't do it in this business.

And so yeah, we do tend to take a view where you have a long-term relationship with your customer. Your customer is sitting there having, in some cases, many employees spend their whole working day putting stuff into your system and storing stuff in your format, and accruing capital. And as long as there is a CAD market, as long as there's an AutoCAD, it's going to read every AutoCAD drawing back to Day One. However different the environment. However different the user interface.

Windows and these other things give applications that sense of immortality. In the sense that once you buy into it, clearly there's going to be an upgrade path. When we get full-motion video integration, when we get 3D viewports of the models, it's going to be done in device-independent way.

I think we need only look at the integration of sound, that you can see both in MIDI Manager on the Apple first and in the Windows Multimedia Extensions. It used to be that if you wanted to do anything with sound, it was all hardware-specific. Anything you did was obsolete in six months because the board that you designed it with wasn't made any more. Even if you had three systems, they were all different, because you simply couldn't buy the parts that you put in the first system, even if you were happy with something that was obsolete.

And now you write something for MIDI Manager on the Mac, you write something for Multimedia Extensions, and you know it's basically going to work for a very long period of time. Because the drivers have been decoupled from the application.

And that's really what these systems do for us as application vendors.

Q: Are there areas you'd like to be in that you aren't?

Oh, hundreds. Hundreds. There's no lack of opportunities in this business. There's just a lack of the time to do anything about them.

Q: These damn finite carbon units....

Well, yeah. It's lack of attention to do it right, I think, that's the most obvious thing. I think we could define 25 products that our existing AutoCAD installed base would love to have that we're getting around to one or two a year at this point. If you look at opening up new markets, there's absolutely no problem coming up with ideas.

Most of them are bad, but again, it doesn't cost very much to find out that they were.

We'd like to do another 35 titles in the science series. We're trying to make our science series like the Time-Life Books science series, except dynamic living things that you can do real things with, rather than presentation in a book that becomes dated. We can only do about one a year right now; you don't find guys like Rudy Rucker standing in line at the unemployment office. [laughs]

So there are those constraints. But there's never any lack of ideas in this business.

Q: One of the good things about Autodesk as a company, it's always seemed to me, is that there's always been the perception at key levels that the company's crown jewels weren't some code sitting on somebody's hard disk somewhere, the crown jewels were in the heads of its people.

Mm-hm.

Q: Do you feel like that's an accurate perception, and if so, how do you sustain it in the face of corporate inertia?

I think, in a software company, you only really have two assets. You have your position in the market and your installed base, and you have what's in the heads of the people who work for and with the company. Because what you're selling today is not going to exist in 36 to 48 months.

I think everybody in this business totally rebuilds their product, even though it has the same name, over that period of time. You don't start from scratch, you simply replace. If you start replacing 20-30% every release, before long you have a totally new product. And what determines what that product is in the heads of the people who are working on those new releases right now.

It is of surpassing importance to make sure that you have the best people you can possibly get. Information Letter 14 was written assuming that people had digested an earlier 25-page memo that never got out that explains this.[419]

The nature of a business is inextricably related to the constraint upon its growth. In other words, what is the constraint that keeps a company from growing faster, doing more, whatever. And that varies from industry to industry. In a capital-intensive industry, it's how much money you have and how much money you can raise. Lots of businesses are like that. Some businesses are natural-resource-limited—how many trees do you own? How many trees can you cut down a year, if you're a paper company.

The software business is constrained by talent. Human talent. How many talented people can you get to work for your company? How many talented people can you develop from people that come into the company and wish to be trained? And that's not just programmers, by any sense. It's people throughout the organization.

You need talent everywhere in the company—it's that broadly based talent, of being engaged in the industry. In the software business, you're not investing heavily—we didn't even own a forklift for the first five or six years. There's no capital equipment. Even the PCs don't cost much—they cost like a guy's salary for a month.

You're investing in people. And if you don't invest in developing the people you have, in getting the right people, you're going to fail. It's just that simple.

This isn't Marin County airhead mush, it's just good business. If your asset is people, and you don't invest in the asset, you're crazy. It's like you're running a railroad and you don't invest in the rails—after a while, the trains start running off. It's just stupid.

So I think you simply have to do that. And if you make that investment, then the company can grow over a period of time and continue to succeed.

I think a lot of the problem that you have in business, and also in politics and so forth which I'm *totally* not interested in, is that it's always easy to neglect maintenance right now. The impact is always deferred well past

[419] I was referring to "The New Technological Corporation." See page 459.

the current horizon, or perhaps even somebody's retirement date.

If the railroad starts to get in trouble, "well, let's just stop fixing the tracks." And you know, for the first three years nothing really bad happens. And then, unfortunately, that guy retires and a new guy comes in, and the trains start running off the rails, and "oh gee, it's going to cost 4 1/2 billion dollars to replace all the tracks on the railroad, and we don't have that much money." And then you're stuck.

Similarly, if you take a typical software company, you can fire all the programmers, and probably for 24 months nothing goes wrong. You make all kinds of money. And then all your competitors come out with new releases and you say "Oops! Guess we should have had a new release. . . ." And then you're stuck.

It's just sound business, in a business that is talent-constrained, to do everything you can to attract and retain the talent that you have. Because that's really all you have.

Q: Given that you can find them, what tends to keep talented people interested and involved? I love your thing about "Make the best product. No bullshit. Reward the people that do the work." But what does that mean in real life?

I think one thing that people don't pick up when they talk about Autodesk—because Autodesk kind of had this image, at least, of being a very far-out-on-the-leading-edge kind of company, in some ways—is that of all the guys who founded Autodesk, we were all maintenance programmers. We weren't Xerox PARC researcher type guys. We were guys that worked on these big operating systems that live for 25 or 30 years, out in the field, fixing bugs.

We were *maintenance* programmers. And the culture of the maintenance programmer is very different than the culture of the research guy. The maintenance programmer works directly with the customer. Maintenance programmers recognize that someday they're going to retire and their children will be maintenance programmers, who are going to have to inherit this thing.

But really, it's more of an ethos that is that of engineering rather than that of a writer or of a moviemaker or something. It's recognizing that the thing has a life cycle, that there are lots of people involved in it, that if you write something and don't adequately document it so somebody can pick it up, you're not doing your job. You're fundamentally not doing your job.

And we try to take that approach to everything we do. We certainly rely on individual talented people, and one of the things that's unique about this business is that there are individuals who are hundreds of times more productive than other people who are productively employed in this business. When you find those superstars, great. But you can't build a company out of all superheroes, because there aren't enough superheroes to go around.

Q: And besides, they don't necessarily get along when you get them.

[laughs] That's right.

If you're fulfilling the needs of real customers, you have to be able to do it with a product cycle that fits the customer's expectations. You have to meet the needs of the marketplace, you have to evolve onto the new hardware that's coming along. And that means, in a large developed market like the market for AutoCAD or for spreadsheets or for word processors, you have large teams of people doing lots of things at once, doing lots of things in parallel. And that's management. You need to have management that understands how to run big engineering projects.

It's not the most exciting thing in the world to talk about, but if you want to be in this business you've got to know how to do it. You can't design and manufacture cars with one bright guy that knows a lot about cars. It takes a lot of people in that cycle, all the way through. And that's the way we have to build software, when software begins to become as complicated as cars.

People say, "Well, wait a minute. A car isn't very complicated." But these are people that don't cut metal. I remember years ago I heard George Morrow speak,[420] and he said, "You know, I thought this computing stuff was really complicated, but you've never really done anything until you've punched a hole in a piece of metal." He was talking about his adventures making the cover for the first one-megabyte disk drive that he sold, and getting all the holes to line up right, then going to a second supplier and they punched the holes in the wrong place.... And recognizing that there's no Undo command once you've punched a hole in a thousand pieces of sheet metal. [laughs]

It's a perspective you have to have. Particularly as the user expectations grow. When you go into Egghead, and pay $35 for a piece of software, you expect to have something that's got a large number of man-years of work in it. I may say your expectations are ridiculous [laughs], but they're yours, and you've got the money, so I better conform.

I think part of what's happened is that the kinds of things we vendors had the audacity of selling for money four and five years ago are what you expect out of a public domain program now. So if we intend to go on asking people to pay money for this stuff, we're going to have to really give them something that's worth it.

And that's a big job. Again, when you talk about the big CAD systems, the mainframe CAD systems, those are systems that are 50 times bigger than AutoCAD. We have a long way to go just in that line, and our users can use every bit of that capability just as soon as we can get it to them.

But those are big engineering projects that have to be done in a rigorously professional manner, or you're going to fail. If you talk about building a system that has conceptual design on the front end, drafting in the middle, manufacturing on the back end, it integrates to your manufacturing control system, it complies with the ISO 9000 specs for quality control that are being enforced around the world now, that communicates to other CAD systems in the PDES product description language that is now becoming standardized—you end up with a couple hundred people working on these things, and all the pieces, and they've got to plug together when you put it in a customer location.

Q: How do you avoid the design-by-committee problem when you do that?

Well, at some level I'm not convinced there's a problem with that. You certainly look for inspiration, but also you have to get there from here. In the engineering world, many things are designed by committees whether you like it or not. The IGES standard—you can't sell a CAD system unless you can read and write IGES—was designed by a committee. You may be on the committee, but all your competitors are on the committee, and government and industry. People are on the committee, and they hand you this thing and then you go and implement it. We don't operate in a domain that affords us unlimited freedom. ASCII was designed by a committee. Unicode was designed by a committee. These are standards that you have to conform to.

When you go to build your own application, I think you avoid that by having those Wild Talents, and by doing a lot of prototyping, getting a feel for how something is really going to look, and then go out to the customers. Companies that aren't out there with the customers are doomed, I think. The customers have a lot more wisdom—when you've got half a million customers, there's a lot more cubic centimeters of neurons firing

[420] At the monthly meeting of the Small Computer Users of Marin—SCUM.

outside the organization than inside the organization. And it's up to you to really be out there communicating with the customers, looking at what they're doing and learning from them.

Q: Some people would have it that being engineering-driven and being customer-driven are diametrically opposed.

I think that's nonsense. I think that any business that isn't interested in meeting the needs of people in the real world is somebody's personal playpen and not a business. When I say meeting the needs, I mean meeting the needs of the people that are currently buying your products and the people that aren't buying them yet, or aren't buying the stuff that you aren't building yet.

You've got to be customer-oriented. I think every business is a service business. I believe that firmly, that fundamentally when you ask somebody to give you money, you have to be giving that person something that is more valuable to them than money.

Think about that. That's the decision you make. Even when you go into the video store, you're saying "Being able to watch *Terminator 2* tonight is worth more to me than having $1.50 in my pocket that I can do anything I want with." Or even waking up tomorrow with $1.50 in my pocket, and then I get to decide "I want Arnold tonight."

That's what you're doing. And in the software business, it's even more of a service business, because when you're buying a piece of software, you're not buying something that's disposable. You're buying, generally, a commitment to something that you expect to be updated and upgraded, that you expect to preserve the investment you make in hammering parts of your life into this thing.

A software business simply has to respond to those expectations.

Q: It seems that a lot of things you're talking about—long-term strategy, diversification, keeping your edge—pretty much flies in the face of the way business is done in the United States.

That's why I don't do business in the United States any more. (laughs)

Q: So do you think there's hope for business in the United States?

I am uninterested and uneducated in the issues of politics.

Q: I was less interested in politics than in what I've heard coming from the mouths of Mitch Kapor and John Roach and others, that the obsession with the quarterly bottom line is killing American business. Do you find that to be true, or not?

I guess that has killed a lot of companies. But I think it's been, perhaps, used as an excuse by other companies for killing themselves in other ways.

Meeting quarterly figures is an expectation of the United States public stock markets. But the responsibility for the long-term destiny of the company does not rest with the Wall Street investors, it rests with the senior management of the company. If the company has an asset value of a billion dollars, say, like Autodesk used to, and its management squanders that by focusing on the next ninety days, they are not doing their job.

There's a point at which you have to go to Wall Street, and say, "We're going to disappoint you. Because if we don't invest in this business, we are not doing our job as management. We're abdicating our fiduciary duty to the shareholders, to invest in the long-term future of this company."

If you look at companies, you will find times when IBM has gone to Wall Street and said, "We are going to turn in disappointing numbers because we need to develop this new technology. We need to build a 16K RAM plant. We need to have this technology—we need to invest in Josephson junctions or whatever we're doing." That's a decision that they make based on the assumption that twenty years from now, thirty years from now, there will still be an IBM, but there won't be if they don't make the investments they need.

You have to manage the business. You have to sometimes make Wall Street happy, and sometimes you have to make them unhappy. But you're not doing your job—where you fail is when you let the Wall Street analysts make your decisions for you. You have to make your decision, and sometimes, if you're a public company, that means you have to go out and take a lot of heat. But it's better to take the heat up front than it is to take the heat down the road, as surely it is coming if you don't make the investments that are necessary.

On balance, certainly I wish we'd never gone public, looking back on it. It would have given us a lot of freedom, and we didn't fundamentally need to go public. When we made the decision in '85, it was the right decision, because in 1985 we had a company that could run just like a couple of weeks if our sales went to zero. And we expected, within a year, to be competing with IBM, AT&T, ComputerVision, and Intergraph all at the same time. And doing that with no money is kind of a sobering prospect.

So from that context, we really had to do it. The fact that in retrospect we never really needed to is not something that you can undo at this point in time.

I think you do have to ask the question "Is the environment that you're in the environment that is most conducive to long-term success?" And for Autodesk, we're not really coupled to the US, because substantially more of our sales come from outside the US than come from the US, and I expect that to continue. The US is less than a third of the global economy, and you should not be doing a huge percentage of your sales in the US. And we don't.

We do more business in Europe than the US. Particularly since our CAD business is very closely tied to the manufacturing business, you expect a large part of your revenue to come from those portions of the world where manufacturing is basically done, and that's not the US right now. Certainly Europe or the Far East are the meccas of manufacturing, and I don't see that trend reversing in the near future. So being strong there is very, very important to us.

But I think you also have to constantly reevaluate where you're going to make your investments, where you're going to do things. Up until last year, we did all of our software development in the United States. Now we have a software center in Neuchatel, where I work, and we're moving a substantial amount of our new software development and localization there. I expect before too many years pass we may also have a software development center in the Far East.

Q: Are there qualitative differences between the European and Far Eastern markets, as to who your main customers are?

Yes. The Far East tends to be much more of a very-large-company, very-large-account market, whereas in both Europe and the US we have been selling in a very bottom-up fashion to individuals and small customers. Not that there aren't a tremendous number of individual customers in those markets—it really isn't five big companies running the whole thing. But being in those big companies is much more important, because they tend to set the standards for the people that subcontract for them, and the rest of the economy.

Q: A lot of vendors go into, say, the Japanese market, where there's about twelve different operating systems running on PCs and have a hard time getting past that fact—"Do I write for the NEC, or

DOS/V, or AX, or...?"

NEC has 70% of the market. That's a pretty easy decision right there. (laughs) NEC is in a stronger position than IBM ever was.

I think—I get nasty after two hours—I think a lot of companies fail because they're arrogant, ignorant American companies that think they can do business in somebody else's culture the way they do business in California. If you go in, and you're trying to sell somebody a word processor, and it doesn't use their character set, you're probably not going to get very far.

Hey, we're just a CAD system vendor, but we got a bunch of guys to sit down and draw 80,000 characters. If you want to play in Taiwan, particularly, you've got to have traditional Chinese. You've got to have your commands, and the vector fonts that draw on the screen for CAD, and it's a big job. Think about what happens when all those little 256-character tables now have 80,000 characters in them, and there's a whole disk full of just fonts, right? It's a real mess, but if you want to play the game, you've got to be in it. You can't sell cars with the steering wheels on the wrong side, you know? [laughs]

You have to make that investment. Typically the way we organize is that there's a country manager for each country, and that country manager is running the business in that country. They decide which of our products they will sell, what price, what distribution channel, how we do our support—they're the president of a company. Autodesk happens to have a headquarters in Sausalito, but Autodesk is an Italian company, and a French company, and a Russian company, and a Swiss company, because the people in those countries run the Autodesk business there, and they basically make the decisions.

They have to do that. I mean, the thought that you could sit in Sausalito and decide what the price and distribution strategy for Thailand is going to be—it's just crazy. Even if you could do it, it would be ten times harder than having somebody there who speaks the language, who knows the dealers, who visits the customers, who knows who the competition is. Because typically when you go into other countries, you're not competing against three other American companies; you're competing against a whole bunch of indigenous products.

Again, you just have to be sensitive to what people want. The European manufacturer does business in a very different way than a lot of American manufacturers do. And if you want to sell to them, you simply have to solve their problems.

New Leadership for Autodesk

Autodesk's long search for a new CEO culminated in the announcement on April 14th, 1992 that Carol Bartz would become Autodesk's new president and chairman. After all the speculation about my supposed covert rôle in Autodesk management, I thought it essential to declare my enthusiastic approval of the board's choice of Carol, and my commitment to support her in the difficult task she had undertaken. I made the same remarks at the official press conference introducing Carol as Autodesk's new CEO, and at the general company meeting two hours later at Marin Veterans' Auditorium.

Remarks for the New CEO Press Conference

April 14th, 1992

by John Walker

Ten years ago, almost to the day, Autodesk was incorporated in the state of California and began the adventure that has, over the last decade, transformed the computer aided design industry and forever changed, for the better, the way millions of designers create tens of millions of products for billions of customers around the globe. I founded Autodesk in April of 1982 because I perceived an unparalleled and unbounded opportunity created by the continued exponential growth in the computer power available to the individual. It was easy to focus on the opportunity in those days; when a company has little money, no paid employees, no products, and no dealers to sell them, there are remarkably few distractions. Like the knowledge that one shall be hanged in the morning, it concentrates the mind wonderfully. Yet however long and hard the road before us might be, I never worried about failing; so large was the opportunity, so numerous were the paths to success. The only way we could fail would be to lack the vision to see where the technological adventure was taking us and the courage to do whatever it takes to play our part in the industry.

"Whatever it takes."[421] These are the three most liberating and empowering words in the world of business. To understand in fullness, to communicate, and to act in the spirit of "whatever it takes" is what divides companies who succeed and, along the way, spawn new industries, from those who shackle their own success through a lack of the will to win and insure their own failure through the renunciation of "risk."

One year ago, almost to the day, I circulated *Information Letter 14*,[422] perhaps one of the most widely-read internal memos of recent years. In it, I tried to sketch the ongoing changes in a desktop software industry now

[421] This phrase was the centerpiece of Joel Voelz' Scientific Modeling presentation at the February 1, 1992 company meeting (see page 718). I was so impressed by the power of his talk that I focused this talk around that phrase.

[422] See page 600.

entering its third generation, define the challenges that rising user expectations posed to successful software companies like Autodesk, and contend that doing "whatever it takes" to remain competitive would require far more imagination and courage than simply continuing along a path laid out almost ten years before.

Some people have asked me, "why did you write such a memo; why did you circulate it so widely." Well, have you ever tried to change the mindset and direction of a global company of more than a thousand people, with a commanding lead in market share, that's highly profitable, by talking only to a few people at the top? Reshaping an organisation to be competitive in a new era is something that has to be done both from the bottom up and the top down. Last year, with *Information Letter 14* I undertook the bottom-up process of education. The 25 new products Autodesk are shipping this year demonstrate what we have already accomplished.

But vision isn't enough; management alone can't do it, and administration of what is falls short when it comes to realising what needst be. To succeed and fulfill the potential created by all our work over the last decade, Autodesk needs strong, decisive, bold, intelligent, perceptive, imaginative and thoroughly professional leadership. That is what Carol Bartz brings to Autodesk now in her rôle as president and chief executive officer. With the announcement today that she has accepted that challenge, my view of the future before Autodesk is one of confidence and optimism; my concerns are laid to rest.

I do not mean to say that the task that Carol now undertakes is easy, simple, or assured of success. Leading Autodesk into its second decade and next generation will require extraordinary talents, indefatigable dedication, and an unquenchable will to win. It is precisely those characteristics that Carol Bartz brings to Autodesk; that is the kind of leader Autodesk needs; it's the kind of person we sought; it's the kind of president we found.

It is my honour to welcome Carol Bartz to Autodesk. Ten years ago, when I became the first president of Autodesk, I knew that as the company grew the demands on the person who led it would increase. Hoping as I did, and as I continue to do, that Autodesk could grow, in time, to be one of the major industrial enterprises of the globe, I realised that leading this company, realising this opportunity, achieving this goal would require an individual with extraordinary qualities. Autodesk has found such a person to lead us into the next decade. Carol Bartz has my total and unqualified support in this endeavour.

Since the press of Autodesk business in Europe makes it impossible for him to be here today, I'd like to relay Volker Kleinn's support for Carol, and his eagerness to work with her in developing Autodesk's business in Europe and Asia over the years to come.[423] Volker's confidence in the destiny of Autodesk has sustained us all in the difficult months behind us, and his strength and wisdom will continue to form the foundation of Autodesk's success in Europe and Asia, the largest global markets for CAD.

Twelve months ago, I asked whether Autodesk's immediate future would, in the end, be considered its final days. I asked that question hoping to spur the changes that would equip Autodesk for the challenges of an industry that's bursting the bounds of its customary definitions. Today, as our products, our vision, our mission, and our new leadership converge to define the Autodesk of the next generation; the next century; there can be no doubt. These are not our final days. These shall be our finest hours.

How could we possibly fail, except by caution born of cowardice, or lassitude stemming from lack of knowledge of a destiny as simple to plot as a straight line on a schoolchild's notebook?

The wave we are riding has scarcely begun to raise itself above the horizon of human history. Centuries from now, schoolchildren will study our age and daydream, as is their right, "to have been there, then." We are here,

[423] On February 1, 1993, Volker Kleinn resigned, citing "differences of opinion over the strategic direction and management of the European operation."

now. The Golden Age of Engineering is not something in our heritage—not the legendary accomplishments of ancestors, but something that is ours to build: today and tomorrow and tomorrow.

Where is all going? What lies upon that distant shore?

Who knows? Who cares?

It is enough to ride the wave.[424]

The wave continues to mount. The challenges grow apace. Over the next thirty years, to meet the needs of our five billion new neighbours on this planet, we shall have to manufacture as many artifacts as all of humanity have created since *Pithecanthropus*.

The human destiny is not constrained by food or fuel or any of the other conventional causes for despair. Our future is defined by our imagination and wisdom. The mission of computer aided design is to amplify those native human faculties to ensure we succeed. The mission of Autodesk is to bring those tools to every designer, in every industry, in every nation. Autodesk is in the very best of hands-the able hands of Carol Bartz. Our future could not be brighter. I shall support her in whatever it takes to achieve it. Onward to the Golden Age of Engineering.

[424]This is a paraphrase of the Arthur C. Clarke quote, one of my very favourite quotations on any topic by any author, which is given in full on page 429.

Farewell, Xanadu

On August 20, 1992, Autodesk announced that it was divesting itself of the Information Systems Division, returning its equity share in Xanadu and AMIX to those respective companies, and terminating its support of their development efforts. This was the first step in focusing the efforts of Autodesk onto the CAD market.

Autodesk Press Release

The following information was announced today, Thursday, August 20, at 1:20 p.m. PDT:

Contact:
Jackie Rae
Vice President Corporate Marketing

AUTODESK HANDS OVER CONTROL OF INFORMATION SYSTEMS DIVISION

August 20, 1992—Sausalito, CA.—Autodesk, Inc. announced today that it will divest itself of two companies that make up its Information Systems Division. The two companies are the American Information Exchange (AMIX) and Xanadu Operating Company.

"The new management team has carefully looked at what Autodesk's business model should be," said Carol Bartz, president, CEO and Chairman of Autodesk. "Both AMIX and Xanadu are important companies with exciting futures, but they do not fit into Autodesk's core business. We wish them the best in their new endeavors and will work with them during a brief transition."

AMIX operates the world's first electronic, online marketplace where customers can buy or sell digitally stored information: software components and tools, documents and scripts, market research and industry analysis, multimedia applications and the like. Additionally, AMIX provides a complete mechanism that allows customers to consult with each other online.

Xanadu is a hypermedia information server. With an array of information management tools that includes links, version management, detectors, endorsements, and access control, Xanadu makes the ideal underpinning upon which to build collaborative and groupware applications.

Autodesk, Inc. develops, markets and supports a family of computer-aided design, engineering, multimedia, and scientific software products for desktop computers and workstations. Autodesk shares are traded on NASDAQ

under the symbol of ACAD™.

AMIX Press Release

The following information was announced today, Thursday, August 20, at 1:20 p.m. PDT:

FOR IMMEDIATE RELEASE

Contact:
Jamie Dinkelacker
AMIX

AMIX ANNOUNCES INDEPENDENT CORPORATE STATUS

August 20, 1992—Mountain View, CA.—The American Information Exchange (AMIX) announced today has spun off from Autodesk, Inc., and formally establish itself as an independent company.

"Autodesk has been a mentor to AMIX for the last four years," said Dana Timbrook, president of AMIX. "Not only have they committed significant resources to the company, but they also provided pivotal guidance and feedback as we developed our online information service and studied the market. Now, with the opening of our first marketplace to the public, we feel we are ready to stand on our own.

"A number of strategic companies and individuals are interested in investing in AMIX and our next challenge is to close a new round of funding and continue our market development activities," Timbrook added. "We see the company's future as very bright and are excited to be moving into another growth phase.

"The new Autodesk management team is focusing very carefully on what the company's business model should be over the next decade," said Carol Bartz, president, CEO and Chairman of Autodesk. "AMIX simply doesn't fit into our fundamental business and this is the basis for our decision to step away.

However, AMIX is a groundbreaking company with much to offer the marketplace. We wish them well as they move into the next phase of their corporate development."

AMIX operates the world's first electronic, online marketplace, where customers can buy or sell digitally stored information: software components and tools, documents and scripts, market research and industry analysis, multimedia applications and the like. Additionally, AMIX provides a complete mechanism that allows customers to conduct consulting with each other online. AMIX handles all accounting and billing functions and has binding arbitration to resolve any possible disputes.

AMIX, now as an independent, private company, recently opened for business with the public and will focus its initial marketing efforts on the software tools and developer communities before expanding into other information marketplaces. AMIX has received interest from the public, tradeshow attendees and the investment community.

AMIX is located at 1881 Landings Drive in Mountain View, CA., and inquiries should be directed to 415-903-1000.

Autodesk Inc. develops, markets and supports a family of computer-aided design engineering, multimedia, and scientific software products for desktop computers and workstations. Autodesk shares are traded on NASDAQ under the symbol of ACAD(TM).

Message from Carol Bartz

Employee Addendum to AMIX and Xanadu Press Release

As the attached press release states, Autodesk has officially given notice that it will sever it ties to AMIX and Xanadu at the end of 30 days. This is being done for a fundamental reason. In developing our strategy for the future we determined that we want to commit our resources exclusively to programs that strengthen our core business. While both AMIX and Xanadu are doing important work, they simply do not fit into Autodesk's core strategy and the management team feels we should devote these resources to future strategic opportunities.

Autodesk is providing both companies with a period of transition in which to locate other funding and move forward with their business plans. And Mark Stiegler, who is an Autodesk employee, has chosen to leave Autodesk and officially join AMIX and Xanadu. Our discussions with both companies, however, indicate that they believe they have real funding sources ready to step forward and plan to continue as independent companies in the future.[425]

I want to assure you that this was done with the utmost sensitivity to each company's particular needs and they responded with appreciation to Autodesk's concern and commitment. I believe, and both management teams from AMIX and Xanadu told me, that they looked upon Autodesk as a mentor over the last four years and understand the business reasons behind today's decision.

If you have concerns or questions about this news, please bring them to the employee meeting on 8/31. I will be happy to address any outstanding issues in that forum. Thank you.

[425] In the spring and summer of 1993, I attempted to make an investment in AMIX to fund its launch as a public Internet service. Concurrently, the state of California filed an absurd and abusive tax audit against me which asserted, even though I'd left the United States for good in May of 1991, that I remained, to this day and forevermore, a resident of California and owed them tax on my income from working in Switzerland. Upon being notified of this action, I terminated negotiations with AMIX, vowing henceforth to take no action of any kind which resulted in a net transfer of my worth into California. AMIX is, at this writing, still seeking capital to re-launch the business.

Micro Engineering Solutions Acquisition

On October 16, 1992, Autodesk announced that it was acquiring Micro Engineering Solutions, Inc., developer of SOLUTION 3000 and other mechanical CAD/CAM products, with the intention of integrating their NURBS-based surface modeling technology into the AutoCAD family. This represented a significant change of direction in Autodesk's mechanical strategy: beforehand, it was envisioned that AME would be extended by integrating the ACIS modeler, with the bulk of the work being done in-house.

In November 1993, AutoSurf Release 2 was announced, which provides the first true integration of the MES modeler into AutoCAD.

Len Rand on the MES Acquisition

Comments from Len Rand, VP of the AutoCAD division, to accompany MES acquisition news release:

"I have great news for you and Autodesk in general. We acquired Micro Engineering Solutions (MES) Inc., (see accompanying news release and Q&A).

Autodesk acquired MES, a leading developer and marketer of manufacturing CAD/CAM software, because we wanted to strengthen our 3D modeling, design, and manufacturing capabilities. One of our main CAD strategies is to become a major player in the MCAD market, a significant and fast-growing market, and MES provides us with a lot of ammunition to take advantage of this tremendous opportunity and really give the competition a run for its money."

Today, Ken Spenser, president and CEO of MES, said: 'People need to recognize this acquisition is more than just a purchase—it represents a commitment to supplying world class MCAD products to customers worldwide.'

Situated in Novi, Michigan—the heart of the U.S. manufacturing industry—MES is a privately held, five-year-old company that is recognized as a quality supplier of MCAD software. The founders of the company are from GM and the relationships they have built in the automotive industry are very strong. In addition, the 50 employees not only share our vision for our future role in MCAD, but they also have a similar culture to ours and will be great to work with.

We welcome this addition to the Autodesk team. MES will continue to work in Michigan, and we invite you to send your warm greetings when possible. On behalf of Autodesk, we are sending MES a videotape—a welcome from the management team—as well as a case of champagne (plus, of course, some T-shirts).

Please join us in making them feel at home here at Autodesk."

Len.

Press Release

The following information was released today, Friday, October 16, at 1:15 p.m. PDT:

For Release at 1:15 p.m. PDT

AUTODESK ACQUIRES MECHANICAL ENGINEERING SOFTWARE COMPANY

Micro Engineering Solutions Joins Ranks of
Leading PC-CAD Software Company

SAUSALITO, Calif.—October 16, 1992—Autodesk Inc. announced today that it acquired Micro Engineering Solutions (MES) Inc., a leading developer and marketer of manufacturing CAD/CAM software. MES, based in Novi, Michigan, provides sophisticated 3D surface modeling and manufacturing technology to various industries, including the automotive sector. Autodesk, with revenues of $285 million, is the sixth-largest PC-software company in the world and the producer of the worldwide *de facto* standard in desktop CAD—AutoCAD(R) software.

"The acquisition brings together two companies who have complementary strengths. Autodesk gains the expertise and technology of a proven performer in the mechanical and manufacturing markets while MES receives the benefits of our established worldwide marketing support and distribution channels," said Carol Bartz, president, CEO, and chairman of Autodesk. "Together, we expect to become a major player in the MCAD marketplace."

Currently, MES product distribution is focused on the company's existing distributor relationships in North America, Germany, and Asia. The two companies are developing a program for increased U.S. distribution and coverage throughout the rest of the world.

"MES's technology is based upon sophisticated manufacturability requirements," said Len Rand, vice president, Autodesk's AutoCAD division. "We will deliver technologically superior engineering solutions that address the entire product development process from conceptualization through manufacturing, emphasizing design-for-manufacturability. This acquisition is the catalyst for Autodesk's Mechanical CAD strategy and will reinforce Autodesk's reputation as the value supplier."

Terms of the acquisition were not disclosed. According to Bartz, the approximately 50 employees of MES—a five-year-old, privately held company—will continue to work out of their Michigan offices.

MES has been a Registered AutoCAD Developer since January 1992. The underlying technologies in both companies' products are complementary.

"This acquisition is an example of Autodesk's ongoing commitment to provide 'best-value' solutions to its customers," said Ken Spenser, president and CEO of MES. "While Autodesk clearly possesses the financial resources to create much of the technology we provide, I believe the company deserves great credit for building its presence in this market by acquiring our proven technology and people that are already closely tied to applications and end users."

"We're excited about the future because of the strong technical fit between our respective product lines and the strong cultural fit between our employees," Spenser continued. "People need to recognize this acquisition is more than just a purchase—it represents a commitment to supplying worldclass 3D CAD/CAM products to customers worldwide."

Currently, through its open-architecture DOS- and UNIX-based SOLUTION 3000(R) product line, MES provides:

- DesignPlus(TM)—3D wireframe design with basic surfaces
- ManufacturingPlus(TM)–3D design, surface modeling, and 2- to 3-axis NC (Numerical Control)
- DesignExpert(TM)—Advanced 3D and surface design and CMM (Coordinate Measuring Machine) data creation
- Manufacturing Expert(TM)—Advanced 3D design and manufacturing, including surface modeling, CMM data creation, and 2- to 5-axis machining
- MILLMASTER(R)—a unique, automated and batch-oriented NC software product for off-hours processing of multi-axis NC information
- CHECKMARK—3D CAD View, Mark, and Check for non technical CAD users, including ability to read/write IGES data (including surface and NC data, make comments, and communicate data electronically)
- A series of translators—a specific strength of SOLUTION 3000–that support IGES, DXF(TM), VDA/FS, DES (GM), CSF (Chrysler), and FST (Ford).

-30-

Q&A On MES Acquisition

The following Q&A is being sent concurrently to all Autodesk and MES employees. Please note that this is an internal document for employees only and should not be distributed outside the company.

Q: What is the news?
A: Autodesk has acquired a software company, Micro Engineering Solutions Inc.

Q: When was this announced?
A: Autodesk management announced the acquisition Friday, October 16, in San Francisco, at a meeting with securities analysts. We also issued a press release, and an email to all employees. Letters and calls to dealers and developers are in progress.

Q: Who is Micro Engineering Solutions (MES)?
A: MES is a 5-year-old manufacturing software company. It is privately held, with approximately 50 employees, headquartered in Novi, Michigan, the heart of North America's manufacturing industry.

Q: What are the terms of the deal?
A: Autodesk acquired MES for just under US$15 million.

Q: What products does MES make?
A: MES provides DOS- and UNIX-based 3D surface modeling and machining software technology. It has shipped over 1400 licenses to companies involved in manufacturing, particularly in the automotive industry. Its products are sold under the SOLUTION 3000(R) product line name.

Q: Why has Autodesk acquired MES?
A: The managements of Autodesk and MES believe an acquisition is in the mutual interest of both companies. Autodesk gains the people, products, expertise, customer base, and other assets of MES while MES joins Autodesk, with our worldwide channel and distribution strength and our proven history of bringing innovative CAD, multimedia, scientific modeling, and other technology to the desktop.

Q: What will happen to current MES management?
A: We believe they are doing an excellent job and we expect them to play key roles in the new organization. They are excited about joining Autodesk and look forward to leading the MCAD market with us in the future.

Q: Do you intend to close or relocate MES operations?
A: Over time our intention is to integrate SOLUTION 3000 development and marketing with that of Autodesk. We fully intend that MES remain intact located in the heart of U.S. manufacturing. Furthermore, development and support of the SOLUTION 3000 product will continue.

Q: How many MES employees are affected? Will there be layoffs?
A: MES has approximately 50 employees. We do not anticipate layoffs.

Q: How will this acquisition impact Autodesk's revenues?
A: We not anticipate a significant immediate impact on revenues.

Q: What impact will this acquisition have on Autodesk's earnings?
A: We expect no impact on earnings—i.e., continued steady, sustained growth—for the next two quarters.

Q: Can SOLUTION 3000 be used in non-automotive applications?
A: Yes. MES's strategy before the Autodesk acquisition was to focus on the automotive and tooling segment. However, SOLUTION 3000 is also marketed to aerospace, consumer products, and defense segments. Autodesk believes that SOLUTION 3000 solves the most difficult modeling and manufacturing problems and is broadly applicable across multiple industry segments.

Q: What percentage of MES's customer base is active outside the automotive industry?
A: Autodesk estimates that 10 to 20 percent of MES's customer base is using MES products for non-automotive applications.

Q: This acquisition would seem to indicate that Autodesk is seeking to compete in the market for manufacturing software. What does Autodesk hope to gain with this acquisition?
A: AutoCAD already has a strong position in mechanical design and drafting with AutoCAD. We believe this acquisition will rapidly expand our position in 3D modeling, design, and manufacturing. It also represents a key step in our longer-term strategy to provide solutions that address the entire mechanical product development process from concept through manufacturing.

Q: Technically speaking, what are the key features of SOLUTION 3000?
A: SOLUTION 3000's key features/strengths include:

- Comprehensive surface modeling

- Surface trimming
- Multiple active viewports
- 3D to 2D associativity
- 2- to 5-axis NC (Numerical Control) machining
- Automatic multiple surface, gouge-free NC machining, including roughing and finishing

Q: What is the competitive positioning of SOLUTION 3000 as compared to products from PTC, SDRC, CV?
A: Autodesk believes MES's technology provides an excellent answer to the needs of customers who need to do 3D design and manufacturing. We believe the acquisition of MES allows us to continue providing market-proven, "best-value" software solutions in the 3D design and manufacturing segment via our strong channels of distribution. Our advantage with SOLUTION 3000 against the major competitors mentioned will include:

- Value for the price
- Superior 3D surface modeling and machining

Q: Won't SOLUTION 3000 compete with AutoCAD?
A: Not really. AutoCAD has a very broad applications base, and is heavily used in many other markets—e.g., AEC, GIS—besides mechanical. SOLUTION 3000 is not only focused on 3D design, it has a specific emphasis on manufacturing industries. In fact, SOLUTION 3000 and AutoCAD are functionally quite complementary.

Q: Does this mean that AutoCAD will be offering some of the features currently in SOLUTION 3000 like surfaces and NC?
A: Our intent is to integrate the underlying technology acquired from MES for 3D surface modeling and manufacturing into the AutoCAD family of products in the future. However there are no specific plans available for doing that at this time, and SOLUTION 3000 will continue as a standalone product line for the foreseeable future.

Q: Will you have a translator available between SOLUTION 3000 and AutoCAD?
A: We're currently evaluating that option.

Q: Are there any planned changes to software and support pricing?
A: For the immediate future, MES product pricing for software and support remains the same as it was prior to the acquisition.

Q: Are foreign-language versions of SOLUTION 3000 available? Are you planning to make international versions available later?
A: MES currently offers Japanese (Kanji) and German language versions. Autodesk will continue to offer these versions of the product. In addition, the SOLUTION 3000 user interface is completely programmable by dealers. As time progresses, other localizations are expected, both by Autodesk and by distributors.

Q: What are the implications of this acquisition for AME (AutoCAD's Advanced Modeling Extension)?
A: AME remains a viable and inexpensive low-end solid modeling product that serves to introduce users to the benefits of solid modeling as well as addressing basic conceptual design and assembly problems. We will continue to sell and support it. In addition, we are exploring various migration paths for users of AME who have a need for the surface modeling and manufacturing capabilities offered by SOLUTION 3000.

Q: How will this acquisition affect Autodesk's relationships with independent developers in the MCAD market?
A: We will continue to work closely with our independent developer community and remain sensitive to their interests and concerns. In the specific case of MES, the SOLUTION 3000 focus on high-end machining leaves ample room for independent development in other segments of that market. We will continue to aggressively pursue core technologies and/or business opportunities that we deem to be in the best interests of our company and our customers.

Q: How will current Autodesk independent application developers who were SOLUTION 3000 competitors (the CAM-only competitors—e.g., CNC Software, Point Control) respond to this news?
A: Like AutoCAD, SOLUTION 3000 is an open system with a strong application programming language called S3L. We hope our independent application developers will take the opportunity to work with us using SOLUTION 3000 as a platform. SOLUTION 3000 does have strong NC capabilities. However, our focus is on integrated CAD/CAM, rather than CAM-only applications. Since SOLUTION 3000 provides solutions for the more complex, multiple-surface machining problems within the manufacturing market, we believe there will continue to be opportunities for Autodesk independent CAM application developers who do not position their products against SOLUTION 3000, and we will continue to work with them in this regard.

Q: Until now, SOLUTION 3000 has been sold directly, with a lot of support from MES. Autodesk mainly sells through indirect channels and counts mainly on its dealers to support its products. How will this difference be handled?
A: We will continue to sell through our highly effective reseller channels, using support from MES salespeople to help ensure the success of our dealers. In addition, we are phasing in a special MCAD dealer training and certification program that is intended to bring on a highly qualified group of AutoCAD mechanical dealers throughout the coming months.

Q: What will happen to MES's current dealers and distributors?
A: Autodesk is currently evaluating how to best transition current MES dealers and distributors into Autodesk. However, the intent is for all current SOLUTION 3000 dealers to continue to carry the product. In addition, Autodesk will make SOLUTION 3000 available to its current mechanical and manufacturing-focused dealer community. For the foreseeable future, SOLUTION 3000 dealers and distributors will continue to work with MES as they have in the past.

Q: SOLUTION 3000 uses HOOPS, and Autodesk owns 20 percent of Ithaca software, the developer of HOOPS. Does this have any implications?
A: Not really. Autodesk invested in Ithaca Software because it is an important graphics standard for use in a variety of software applications. MES made the decision several years ago to use HOOPS because of the advantages of portability which HOOPS provides. MES made this decision independently of the current acquisition and Autodesk relationship.

Patent Nonsense

Ever since Autodesk had to pay $25,000 to "license" a patent which claimed the invention of XOR-draw for screen cursors (the patent was filed years *after everybody in computer graphics was already using that trick), at the risk of delaying or cancelling our Initial Public Offering in 1985, I've been convinced that software patents are not only a terrible idea, but one of the principal threats to the software industry. As I write this introduction in 1993, the multimedia industry is shuddering at the prospect of paying royalties on every product they make, because a small company in California has obtained an absurdly broad patent on concepts that were widely discussed and implemented experimentally more than 20 years earlier.*

Today, software companies are patenting everything in sight, purely to assemble an arsenal to counter-sue anybody who sues them for infringement. Oracle Corporation has taken a public stand against the patentability of software and has forsworn use of its own patents except to counter-sue in infringement claims. The League for Programming Freedom is working for legislation and/or rule-making which would declare algorithms and software non-patentable. Despite these efforts, the trend toward increased litigation, constraining innovation in the software industry, is accelerating. The U.S. government is using trade negotiations to force other countries to institute software patents in their own markets.

While eliminating software patents would be the best solution, changing the law takes a long time and is uncertain to succeed. I've been trying to puzzle out how the software industry might rescue itself from immolation through litigation and came up with the following proposal.

PATO: Collective Security
In the Age of Software Patents

by John Walker
May 10th, 1993

One idea that's been rattling around my skull for the last month or so regarding the patent mess contemplates an active defensive posture. Whether Amurrcan business could (legally) or would (courage/imagination constrained) do such a thing is another matter, but in essence it's a 1990s replay of how the radio companies untangled their patent mess in the 1920s by cross-licensing all their patents through the newly-formed RCA.

Basically, I've been thinking about using NATO as a model of a patent defence consortium. Suppose a bunch of big software companies (perhaps led by Oracle, who's already taken the point on this) were to form PATO—-Patent And Technology Organisation—and contribute all their current software patents, and all new software

patents they were granted as long as they remained a member of PATO, to its "cross-licensing pool". To keep the lawyers and shareholders from going nuts, the patents would be licensed through PATO but would remain the property of the member—a member could withdraw with appropriate notice and take the patents back from the pool.

Any member of PATO would be granted an automatic, royalty-free license to use any patent in the cross-licensing pool. Thus, by putting your patents in the pool, you obtain access to all the others automatically (but if you withdraw and pull your patents, of course you then become vulnerable for those you've used, which creates a powerful disincentive to quit).

The basic principle of NATO is that an attack on any member is considered an attack on all members. In PATO it works like this—if any member of PATO is alleged with infringement of a software patent by a non-member, then that member may counter-sue the attacker based on infringement of *any* patent in the PATO cross-licensing pool, regardless of what member contributed it. Once a load of companies and patents are in the pool, this will be a deterrent equivalent to a couple thousand MIRVs in silos—odds are that any potential plaintiff will be more vulnerable to 10 or 20 PATO patents than the PATO member is to one patent from the aggressor. Perhaps the suit will just be dropped and the bad guy will decide to join PATO....

Since PATO is chartered to promote the free exchange and licensing of software patents, members do not seek revenue from their software patents—only mutual security. Thus, *anybody* can join PATO, even individual programmers who do not have a patent to contribute to the pool—they need only pay the nominal yearly dues and adhere to the treaty—that any software patents they are granted will go in the pool and that they will not sue any other PATO member for infringement of a software patent.

PATO is purely for *software* patents—presumably a patent office category can be used to make the definition precise. Joining PATO does not compromise hardware or other non-software patents held by a member.

Now what I really like about PATO vs. many of the other proposals I've seen is that I think it has at least a shred of a chance of being accepted by Amurrcan software companies because, at the heart, it requires absolutely no change in the way they're currently operating. Given how difficult it is to get our leading-edge high-tech business leaders to do anything off the beaten path, this is an advantage. Consider:

Today, most software companies are patenting everything in sight purely to cover their asses, without the least intention of enforcing their patents or looking at them as a source of revenue. When sued, they try to settle based on cross-licensing of something in their arsenal with the plaintiff's patent. With PATO, *nothing changes*—except by joining you're automatically safe against being sued by any other member, and you gain access to a patent pool perhaps 10 or 20 times larger than you own which you can use to beat off any patent weasel who shows up at your door. I've heard talk of ideas like a "patent legal defence fund", etc., but I've deliberately left that out of PATO—a member goes on using their existing patent attorneys for getting patents and defence—thus no change of responsibility and no difficult questions about how much members should contribute to the fund based on size, number of patents, etc.

So the only "big gulp" in joining PATO is renouncing revenue from your software patents. But what big software company is going around suing people today for patent revenue (as opposed to copyright, look-and-feel, etc., which are *not* the domain of PATO)? And in any case, a member can simply quit if they change their mind, getting all their patents back.

It seems to me that if, say, three or four of the top 10 software companies became charter members of PATO, the thing would snowball quickly to encompass the whole industry. I mean, if Joe Software, busy writing the killer app of the 90's in his basement, can, for US$25/year join PATO and thus be protected against patent

claims from, say, Apple, Oracle, Microsoft, Autodesk, Symantec, and Borland, won't he?

Programs Are Programs

The first ten years of Autodesk's existence have witnessed not only a breathtaking increases in the power of personal computers and the capability of the software they run, but equally rapid shifts in how such products are marketed, sold, and supported. There's no reason to believe that evolution in any of these domains will cease in the near future. Indeed, developments in communication technology permit entirely new approaches to software distribution which, being beneficial both to vendors and customers, may be widely adopted in the next several years. In June 1993 I tried to peer into the crystal ball and forecast how software will be bought and delivered in the late 1990's.

Programs Are Programs
How To Make Money In the Software Business
by John Walker
June 14th, 1993

World War Four

In late 1989 I spent some time thinking about what was then called the "emerging new world order", and began to rough out a think piece titled "World War Four". I scribbled an introduction in January of 1990, but since I'd already discussed the gist of the message verbally with just about everybody who was likely to read such a paper, I put it aside and never completed it.

The essence of "World War Four" was the argument that amidst all the optimism and triumph engendered by the collapse of communism—the end of World War Three (the Cold War, 1945–1990), we were in all likelihood at the threshold of World War Four (1990–????). World War Three, a relatively bloodless war which nonetheless consumed far more wealth than any of its predecessors, was about which economic and political system was appropriate for large industrialised nations. That's pretty much been settled now. World War Four was, I predicted, going to be centred on the essential definition of a nation and the appropriate size and scale of political entities. I went on to argue that 45 years of bipolar confrontation had simply put a lid on these issues, which have been at the heart of the overwhelming majority of all modern wars, and that we were probably entering into an era where borders were going to be redrawn all around the world.

Well, of course, if I'd finished the piece and managed to get it published anywhere, I'd probably be spending all my time on the talk shows as the Political Prophet of the Nineties, so it's just as well I didn't. But I stand by my 1990 predictions, including the one that there will be a serious secession movement by one or more

states of the U.S. before the end of this decade.

So what does this have to do with the title of *this* paper? Well, nothing really other than the fact that I've put everything aside to finish this paper because I have the feeling that if I don't it's going to suffer the same awful fate as "World War Four"—namely come true and be considered totally obvious before I get around to predicting it. I've been developing the ideas in this paper over the last 14 months—ever since the idea popped into my head during a conversation with Bill Gates—and as I've explored it and thought further it seems more and more compelling. I know for a fact that Gates is thinking in this direction as well, because I asked him and he said, "Yes". Until recently I didn't think, however, he had the whole idea put together as cleanly as I did. Recent events make me suspect he's way ahead of everybody.

As with "World War Four", I've discussed aspects of this with many of you. Because I've tended to focus on one aspect or another of the whole picture, and because my views have been evolving, in part due to your valuable comments, I'd urge you to read this document anyway to make sure you see how all the pieces fit together.

Background—Is Software a Business?

In a world where Microsoft is worth more than IBM, it's hard to imagine that less than 10 years ago a substantial part of the financial and venture capital community were skeptical of the fundamental viability of the software business—in fact it was fashionable to ask whether software was really a "business" at all, in the conventional sense.

Certainly, in 1983 and 1984, one could point to Microsoft, Lotus, and Ashton-Tate as profitable, rapidly growing companies, but rare was the company which broadened an initial, seemingly random success into a consistently successful product line. Further, a software company seemed threatened from all sides—from piracy, from next-to-zero-price competition, and from an unsettled distribution environment. Rarely was software even mentioned in reviews of trade shows, and people like me who argued that software drove hardware successes—that the Apple 2 was successful *because* of Visi-Calc and the IBM PC *because* of Lotus 1-2-3 were considered more than a little daft.

And yet today, in a world where software companies post profit margins unheard of in most legitimate businesses, the old doubt about the viability of the software business remains, but in another guise. In 1983 they asked, "Can you really make money, long term, selling software?". In 1993, they ask, "Can anybody other than Microsoft make money, long term, selling software?".

This is not a facetious question; when company after company which were once pillars of the industry: Ashton-Tate, Software Publishing, Borland, one after another collapses, is gobbled up, or is on the ropes against deadly price-cutting competition, one cannot help but ask whether any software company can truly consider its position secure as long as there is a well funded competitor willing to sell at close to the marginal cost of goods in order to gain market share.

I believe we are in the midst of a fundamental shift in the way software is distributed—a transformation in the relationship between software vendors and their customers fully as significant as the emergence of computer retailing, spurred by the nascent desktop computer. Companies which anticipate this transition, who encourage it and prepare to benefit from it, may find themselves in as unassailable a market position as any company in this century—the "natural monopolies" of the next.

It's a Program, Stupid

How odd it is, that every day we use the word "program" to talk about the products we create and sell in the software business, and also to describe the information and entertainment we watch on the television and hear on the radio. And even though the word "software" has become commonly used to describe video cassettes, compact discs, and other non-computer material, rarely do software executives see a link between the "programs" they sell and the "programs" they watch. Many managers in the computer software business, and most analysts who follow it, came from a hardware or turnkey systems background; thus it is inevitable that they often think of software as a product like a television, rather than an ongoing service like the "programs" the television delivers.

Let's Crunch Some Numbers

Okay, I'm going to jump way ahead of myself here in a brazen attempt to grab your attention. If you agree with this analysis, I'm sure you'll summon the strength to trudge through the more deliberate development of the argument that follows.

It's 1997. You call up the 800 number to order another computer, and after you've chosen between the Alpha-III and the Octium chip and the 15 and 30 gigabyte hard drive, the salesperson tells you that the machine comes with the "Basic Package" of Windows NT, Word, Excel, Access, Money, and Multimedia Producer, and asks if you'd like to turn on any additional software at the time. You request Project, Designer, and Visual C++, and they're enabled also. In any case, you're told, "it's all on the CD-ROM, so you don't have to decide right now".

The computer shows up, and you start using it. Late one night you decide you really need the German language spelling checker add-on for Word, so you put in the CD-ROM, call Microsoft's 800 number to order it on your credit card, and get back the code you enter to turn it on in your smart card. Or if your computer has a modem, you can do it just by clicking the mouse.

Basic costs you $10 per month. "Premium" programs range from $.75 to as much as $100 per month for exotic niche applications. Every month you get your "Microsoft Bill" itemising everything you subscribe to and what it costs. Most folks just have it paid automatically from their credit card, generally 6 months at a time. If you don't pay, you don't get the new authorisations for your smart card and the program stops working. So you pay. Every now and then a new CD-ROM shows up with all the latest updates and upgrades and new products, each with its "try me, buy me!" demo you can run right away. As long as you have Basic, the CD-ROMs come automatically in the mail every 3 months.

Now let's look into the other end of the binoculars; from Bill Gates' chair rather than his customers'. Today, there more than 125 million MS-DOS personal computers installed. Given the rapid adoption of Windows and sustained high sales rate of new machines driven by price performance improvements in new chips, I believe it conservative to expect that 100 million Windows NT machines will be installed 4 years from today, most equipped with CD-ROM, multimedia accessories, and contemporary peripherals; some upgraded from current high-end MS-DOS machines, but most new machines of the Pentium/Alpha generation and their successors. Further, let us assume that Microsoft is unsuccessful in selling *any* software other than the Basic set (I'm sure you'll concede, based on Microsoft's new product success rate, this assumption is conservative). Well, multiply it out. That's 100 million machines times US$10 per month times 12 months per year, and the answer is:

US$12 Billion-with-a-B-like-Bill per year of *automatic recurring revenue* for which the marketing costs are essentially nil and distribution margin is nonexistent since fulfillment is direct.

Now given an utterly reliable, competitively unassailable annual revenue stream of US$12 billion per year, you can invest in fundamental and applied research, technology development, new product development, marketing, and launch at levels no other player can approach. These investments translate directly into additional recurring revenue to the extent the premium products are adopted by the 100 million and growing installed base, and the proceeds fund further development. In this environment, competitors are forced to either cut prices (and thus their margins), or search for a genuine technological edge and rush it to market before the folks who employ more than 50

Does This Sound Familiar? 1

Well, Walker's gone right off the deep end again, without even bothering to fill the pool this time? Smart cards, CD-ROMs, expiration of programs, blah, blah, blah. He's made up a whole fairy castle industry out of thin air without the slightest proof that it could even be viable.

And yet, there is something about it that seems oddly familiar.... Let's see.... Aha!!!

Programs are programs. So let's start by looking at what's on the television. No, not the pap on the screen, what's *on* the television—not the bloody penguin but the little black box the penguin's standing on. Today, not far out in the distant 1997 I was talking about, more than a hundred million people in Europe and North America buy programs—television programs—on a monthly basis. They buy them from a cable television operator or, if they live outside an area with cable service or wish a wider selection, by subscription to a satellite broadcasting system. (Satellite broadcasting still seems a little exotic in the U.S., though that is rapidly changing with the Hughes-Thomson-RCA DirecTV system; In Europe it's everywhere—it's hard to find a home in Britain without an Astra dish—satellites work better in Europe and Japan because they aren't as *big* as the U.S.—a 2 foot dish works just fine, and you can buy the whole rig for about US$300). If you use a satellite dish, your receiver has a little slot where you put in a smart card. If you decide, for example, to subscribe to Turner All-Colourised Movies, just pick up the phone, call the toll-free number, give your subscriber ID from the smartcard, and zap-flash in 30 seconds you're watching Bogey in living — well — pasty colour. This technology is off-the-shelf stuff available in every Radio Shack in the U.S. and any T.V. store in Europe.

Most cable television subscribers pay US$5 to US$10 per month for Basic and monthly fees for Premium services like:

Low rent cable channels (A&E, BRAVO, etc.)	US$0.79 - 1.00/month
Network packages (Denver 5 or Primetime 24)	US$4.00 - 5.00/month
Premium Channels (Disney, HBO, TMC, etc.)	US$7.00 - 10.00/month

and the typical satellite user (who receives all the channels included in Basic cable for free) pays between US$150 and US$300 per year for premium services. Of course there are ultra-premium niche services such as real-time stock and commodity quotes, etc., for which one may pay up to US$100 per month.

You do not *buy* your television programs, you *subscribe* to them, and the revenue flows back through the chain to those who manufacture them (have you noticed how often you see "An HBO Picture" in the titles in the theatres?). And if folks pay $150 a year or more for television programs, is it absurd to suppose they will

pay $120 a year for computer programs, especially when the cost is in little monthly nibbles rather than $495 up-front the way we do it today, and when you can always rationalise a purchase by saying, "Well, I can always cancel it if I don't like it"?

I believe that soon we're not going to *buy* computer programs either, we're going to *subscribe* to them. Programs are programs.

Does This Sound Familiar? 2

What companies stand out as the huge unassailable (for a while) monoliths of this century? In the United States I'd list:

American Telephone & Telegraph	1875–1980
IBM	1930–1970
Xerox	1960–1970

Now consider that during the time that each of these companies was in a position of total dominance of its market, it delivered its product, which was fundamentally a piece of hardware, as a *service*, almost entirely on a rental or subscription basis. This, combined with a dominant market share obtained either by getting there first (AT&T/Xerox) or by blowing away less-serious competitors with a massive sales organisation (IBM in computers after 1948), largely insulated the base revenue stream of these companies from business cycles and competitive threats. The annuity-like revenue base, in turn, allowed them to make large, long-term investments in technology relevant to their business (Bell Labs, IBM Research Labs, Xerox PARC) and in product development aimed at further distancing them from their competitors.

Note that in each of these cases the subscription/rental nature of the revenue stream allowed these companies to subordinate technological progress to the needs of the business. Unlike a free-for-all like today's RAM chip or hard disc market, where product generation times are measured in months, AT&T could introduce direct dialing, direct long distance dialing, electronic switching systems, etc. on a decades-long plan geared to optimising their profits. IBM was not forced to rush out the 7094 or 360 under the gun by competitive fears—they could switch their rental base to a new generation at a time of IBM's choosing, when the technology was ripe to increase their revenues and earnings. In short, when a company achieves a stable subscription base, *it* calls the technological shots in the market. Of course if a company is complacent, it will eventually be knocked out, but you have to be awfully complacent and/or incompetent to nullify the benefit of a 10 to 1 advantage in product development and marketing resources (Xerox, of course, demonstrates that it *can* be done, but in the other cases it took government action or fear of government action to displace the dominant player).

Does This Sound Familiar? 3

Redmond, Washington, March 20, 1992. Transcript (from memory) of meeting of John Walker and John Forbes of Autodesk and Bill Gates and Todd Needham of Microsoft.

Walker: So let me see if I understand where you're going with this, Bill. What you'd really like is if in, say, five years, everybody with a computer gets a Microsoft bill every month, just like a telephone bill, for each product they use.

Gates: Precisely.

Does This Sound Familiar? 4

Have you heard of the "Microsoft Developer Network"? I'm a member. Walk around and ask random programmers if they are as well (either company-paid or on their own account). There are 45,000 members as of the last time I looked, and I suspect the ranks are swelling rapidly. If you're a remotely serious Windows developer, you simply cannot afford not to join, because it's the only effective way to receive massive amounts of source code, internal technical documentation, beta copies of soon to be released Microsoft products, special development, debugging, and authoring tools, etc., etc.

How does it work? Well, it's a, er, *subscription*. I mail in a check for US$200 per year, and every 90 days they send me this, er, CD-ROM filled with 600 megabytes of ever-changing goodies. And as long as my subscription is current, the bits just keep on coming.

Does This Sound Familiar? 5

Have you noticed how Microsoft update marketing works these days? They've pretty much dropped even the pretense of running updates through the distribution channel except for mega-million blowouts like Windows 3.1 or DOS 6. No, as long as you've registered the product, right about the time the new release hits the cover of PC World, a little letter from Bill shows up with their little blue OCR form. You just tick the box and attach your check or credit card number for a readily-digestible fee (usually US$30 through US$150) and bung it in the post. A week later the update shows up at your door. For those who update faithfully this way (and I won't get into the many ways Microsoft forces you to stay current or pay a heavy price—they're masters at it), this amounts almost to a subscription—not on a consolidated basis like cable or satellite TV, but to a product, like a magazine subscription. And the economic dynamics strongly resemble those of magazine subscriptions.

Magazine subscriptions are typically sold through retail channels and/or highly discounted for an initial subscription; publishers don't usually make money on initial subscriptions. Renewals, however, are handled by direct marketing and all the recurring revenue for the subsequent years goes right into the publisher's pocket. It's the same for Microsoft Word; Gates has every incentive to discount initial sales of Word as much as possible to gain market share against Word Perfect, and to grant all kinds of incentives to his channels as long as he believes that each initial sale plugs him into a virtually guaranteed revenue stream of, say, $50 per copy per year. Of course to realise that revenue he has to keep the updates coming and provide enough added value in each one so people continue to stay current. But since I've never seen a wish list get shorter for any software product I've worked on, I hardly think that's a problem as long as you're willing to invest in development.

Does This Sound Familiar? 6

Microsoft, 2 Cable Giants Weigh Interactive TV Venture

By John Markoff
New York Times Service
International Herald Tribune, Monday, June 14, 1993

> NEW YORK—Three dominant technology and entertainment companies are on the verge of joining forces to create the equivalent of software for cable television—a system that would combine the worlds of computing and television and perhaps shape how much of popular culture is delivered.
>
> Time Warner Inc., the largest entertainment company, Tele-Communications Inc., the largest cable television company, and Microsoft Corp., the largest software company are expected to announce by the end of the month that they will form a company, tentatively called Cablesoft. The companies hope the new venture will lead the way in establishing a standard for the transmission of a coming generation of interactive programs.
>
> At stake is control of the unobtrusive cable box that sits atop many television sets. In recent months the box has become a battleground for computer, telephone, and cable companies.
>
> . . .
>
> Last month, for example, Intel Corp., the world's largest chipmaker, Microsoft, and General Instrument Corp. announced plans to develop a cable converter that would have a built-in personal computer. Last Monday Time Warner announced that Silicon Graphics Inc., a Silicon Valley computer maker, and Scientific Atlanta, a supplier of cable boxes, would supply hardware and software for its digital television trial in Orlando, Florida, which is scheduled for next year. A day before that announcement, Kaleida, a joint venture of IBM and Apple Computer, said it was joining with Motorola Inc. and Scientific Atlanta to develop a similar futuristic television controller.
>
> . . .

Need I point out that just as soon as your cable box talks to your computer in almost any fashion whatsoever, the technological means exist to make subscription software as secure as subscription television. And the pirate TV decoder business seems to have been roundly wiped out.

The Time Warner/Tele-Communications/Microsoft deal is, of course, something that's looking out a few years and probably genuinely focused on interactive television. But purely as a side effect, something that just falls out, is the means to distribute and authorise subscription software world-wide. Controlled by Microsoft. This could have implications outside the entertainment world.

Getting There

Ever since 1985 when I first I proposed the "AutoCAD Professional Subscription" as a means of finding the holy grail of recurring revenue, I have been following the evolution of the software business from a retail sales model to a subscription/service base. When I look at the convergence of the trends and events I've noted above, I cannot help but believe we are, if not already in that era, at least on its threshold.

Emerging from this period of transition with a large, stable, and growing subscription base will render the companies who succeed formidable, almost invincible, competitors compared to firms with smaller market

share forced to generate their revenue entirely from new sales. Reinvestment of a stable revenue base can, if done wisely, further widen the gap between the dominant firm and the dwarves.

For some reason, when I discuss the subscription model of software distribution, many people get confused and think I'm talking about something in the medium to far future—"Well, yes, things may indeed go that way once we have interactive television/fiber to the home/data highway/..., but for now....". But other than the Microsoft cable box venture, which is interesting but unnecessary, nothing I have discussed has any technological contents whatsoever; it is purely a question of marketing and distribution strategy. Bucks, not Buck Rogers.

To summarise and demonstrate that all the foundation pieces exist:

- Tens of thousands of people pay Microsoft US$200 a year to subscribe to Developer Network and receive a CD-ROM every 90 days.
- Silicon Graphics distributes all its software to every customer on regularly-issued CD-ROMs. You purchase an authorisation code to install and enable it.
- Microsoft does direct marketing and fulfillment of most updates of current products.
- More than 50 million cable television subscribers pay US$100 per year, some far more, for their television programs.

Unlike many major transitions in distribution strategies, moving toward a subscription model can be done, as far as I can tell, with little or no risk (effort and expense, yes; risk, no). In a business which concentrates primarily on new sales, the installed base is often an underperforming asset waiting to be discovered. Moving to direct marketing of "frequent, cheap" updates and upgrades, as Microsoft has done, is unlikely to alienate existing channels geared to selling new products. As Autodesk has discovered, as long as we keep new sales of AutoCAD firmly in the dealer channel, providing direct options for "the little stuff" may provoke grumbling, but seldom more grumbling than we hear about "unprofitable, time wasting update business".

I believe a subscription strategy can be evaluated and planned relatively simply once we discover the answer to the following question:

> How much revenue do we generate, per annum, from the average unit of AutoCAD, after its sale?

Now, I don't have the vaguest idea of this number, but let's play a little napkin engineering and make a wild stab. The wild and wooly R12 update generated $22 million in update revenue and the subsequent quarter $13.5 million (Pru-Bache report, May 24, 1993). Let's assume we hold the $13.5 million level (which Laura and folks don't expect, but I want to err on the high side). So we have $63 million in update revenue, liberally construed, in a year with a blockbuster update. Folding the napkin and continuing, we have about a million installed copies, but let's say they're, oh, 700,000 "active" copies, disregarding shelfware and people who haven't upgraded since Version 2.6. Well, that comes out to about US$90 per year per "active" copy. So, for example, if we could get half our "active" users to subscribe for, say US$250 per year, we would have a recurring revenue stream greater than our biggest update year ever, and without all the push and cost it takes us to launch an update. And given what Autodesk usually charges for updates, many users would probably consider this a bargain, particularly if it avoided all the hassle currently involved in updating a copy of AutoCAD.

It's also intriguing to divide the Pru-Bache FY 94 estimate of US$430 million by my "active base" of 700,000. That comes out to US$614 per active unit per year. So were we, for example, to move to a subscription for

AutoCAD of about US$100 per month for new sales, we would generate, month after month, year after year, revenue equal to what we largely derive today only from new sales—again assuming only 50radical change in the way we do business which could be deferred until experience with the installed base upgrade/update program confirmed its viability, or simply put off forever, retaining different channels for first sale and subscription as in the magazine business and Microsoft's current practice.

The Enabling Prerequisite

If you've made it all the way through my arguments without either getting lost in my thorny prose or storming away in violent disagreement with my premises or deductions, you may be beginning to think I'm onto something here. But that raises the legitimate question, "Well, if even Walker's figured it out, why isn't everybody in the industry already doing it?". Indeed, I've wondered quite a bit about that myself.

I think the answer lies in the observation that most companies who succeed in building self-sustaining subscription-based businesses start from a position of effective monopoly of their sector. In the case of AT&T, it was a combination of technology, patents, and government grants which conferred the monopoly. IBM built its first monopoly in tabulating equipment on the patent of the Hollerith card, then clawed its way to an effective monopoly in computers by out marketing and out-customer-servicing Remington Rand, Burroughs, and others. Xerox derived its monopoly from the patent on xerography.

Quite simply, to derive enough revenue from a subscription strategy to make the business run, you have to have the lion's share of the market, not a small slice. To get people to subscribe, you have to have demonstrated technological leadership that convinces them they'll get more value by paying you regularly than buying from somebody else outright, then replacing the product later on. And of course the central development engine needs to be big enough to keep generating the value that gives subscribers value for their money, year in and year out.

Which means that to pull off the transition to a subscription base, you have to start with a large market share lead, and therefore the only companies in the software business well-positioned to do this today are:

Microsoft
Autodesk (in CAD)
Word Perfect (in word processing, but slipping)
Lotus (in spreadsheets, perhaps, and slipping quickly)

Summary and Conclusion

Is software a business? Can a company make money selling software, consistently and reliably? In 1983 many people doubted that these statements were true. In 1993, we have one great success story but little confidence that other sectors and companies are safe investments for the long term with predictable chances for growth.

By 2003, I believe that everybody will know the answers to these questions: "Yes, and yes". Within 10 years the software industry will have restructured itself from a costly and unpredictable bookstore/appliance dealer sale-oriented model to a cable TV-like subscription model. The companies who emerge from the turbulence of

this transition will be the colossi of the industry, no more and no less inherently risky than television networks, book publishers, or regional telephone companies. Their revenues, measured in the billions to tens of billions will fund ongoing product development aimed and increasing their subscription base.

Ironically, they may cease to be viewed as "growth stocks"—once the constant revenue base comes to eclipse the near-term potential of a new product launches, their performance may appeal to those who buy utility stocks today. (But then when electricity use was growing exponentially in the early '50s, utilities were "growth stocks".)

It's 2003. The little black box on the top of the TV is hardly big enough to hold all the logos printed on it: Microsoft, Time-Warner, Swiss PTT, SES/ASTRA, General Instrument, Dolby Labs, Intel, Motorola/Iridium, etc. A couple of wires hook it to the TV and the computer, and the ubiquitous smart card sticks out the front. Every month I get a bill for the programs I subscribe to:

Astra Basic pack	2.00
Eutelsat Basic pack	2.00
HBO	7.00
The Dinosaur Channel	0.50
Canal Plus	1.25
TeleCine Romande	4.00
CNN International	0.75
Microsoft Basic pack	4.00
Microsoft Developer Network	10.00
Microsoft Project	2.00
Microsoft Visual C+++++	9.00
Microsoft Producer	4.00
Autodesk AutoCAD	75.00
Autodesk Cyberspace Explorer	8.00
EuroFeed Internet News Link	1.00
BBC World Service Radio	0.25
. . . about 20 more	. . . programs are programs

And since the bill gets paid automatically, and it's only about 200 francs a month, I don't look at it too closely (other than that AutoCAD thang—wonder if they'll *ever* give me a break!). Each little nibble is so small, though, compared to when I had to shell out US$400 over the counter for some software I didn't know would even work when I installed it, that I pay and pay and pay.

This document is a work of education, not advocacy. I believe that wisdom consists of embracing change, not battling it. We win by enlisting the slow but inevitable forces of economics on our side, not by shoveling sand to halt the tide. All the evidence I can see convinces me the software business is finally making the transition to a recurring revenue model which will be its salvation. This transition will entrench those companies who leverage their large existing market share into a consistent and reliable base of recurring subscription revenue.

Among major software companies, Microsoft and Autodesk (in the CAD sector) are uniquely positioned to lead this transition and benefit from it.

AutoCAD LT

In November 1993, Autodesk announced "AutoCAD LT", which had been developed under the code-name "Madison." AutoCAD LT, priced at $495, became the first product in the company's history priced below $1000 to bear the name "AutoCAD." The specifications of AutoCAD LT invite comparison with the original concept of an "AutoCAD Lite" penned more than nine years earlier (see page 240), and the specifications of the "Windows Engineer" envisioned in the "Nightmare Scenario" of Information Letter 14 (see page 626).

Press Release

AUTODESK LAUNCHES POWERFUL LOW-COST
WINDOWS CAD PACKAGE
New AutoCAD LT Addresses Market Needs

SAN FRANCISCO, Calif.[November 2, 1993] At a press conference today, Autodesk, Inc. introduced AutoCAD LT, a powerful, low-cost, Windows-based computer- aided design (CAD) package, offering a wide range of 2D and basic 3D drafting capabilities. AutoCAD LT software provides full data compatibility and seamless interoperability with AutoCAD software, the world's leading desktop CAD package.

"The advent of Windows makes CAD available to a broad new range of users. AutoCAD LT is ideally suited for architects, designers, engineers, field professionals and managers who need a powerful, standalone CAD tool, but not the advanced features, or robust development platform of a more comprehensive package like AutoCAD," said John Lynch, vice president, Product Development Group, Autodesk.

"Today Autodesk has more than 900,000 AutoCAD users with over one billion drawings, making AutoCAD and the AutoCAD **.DWG** file format a worldwide standard.

With AutoCAD LT, designers and engineers across many disciplines can take advantage of CAD and the AutoCAD design community more quickly and easily than ever before. And they can do it cost effectively," added Lynch.

AutoCAD LT provides a wide range of design and drafting tools, including entity editing and hidden line removal. In addition, with its native **.DWG** file format, AutoCAD LT conforms to the AutoCAD standard and provides users with a natural migration path to the advanced capabilities of AutoCAD software.

"For new users, the Windows interface and simplified command structure of AutoCAD LT help shorten the

learning curve to quickly bring the power of CAD within reach," said Cary Fulbright, director of Autodesk CAD product marketing.

"With its networking capabilities, AutoCAD LT is also perfect for designers and drafters who support the design process in large installations. For example, at existing AutoCAD sites, drafters using advanced AutoCAD features can easily exchange drawings with engineers and managers who just need a basic, yet fully interoperable CAD product like AutoCAD LT to monitor the design process."

Key AutoCAD LT Features and Benefits

- Microsoft Windows Environment: Pull-down menus, accelerator keys, dialog boxes and icon-based objects make AutoCAD LT an easy CAD package to learn and use. The new product supports the Windows Clipboard, as well as Object Linking and Embedding (OLE), which allows users to link AutoCAD LT drawings to other Windows applications such as Microsoft Word for Windows or Excel.

- 2D CAD Capabilities: AutoCAD LT offers many of the same 2D design and drafting features in AutoCAD Release 12. These powerful AutoCAD features allow architects, engineers, designers and managers to create accurate 2D drawings.

- Basic 3D CAD Features: AutoCAD LT includes 2D extrusions, 3D line creation, hidden line removal, shade, multiple views (side, elevation) and the ability to view and edit 3D drawings in the AutoCAD **`.DWG`** drawing file format.

- Complete AutoCAD Data Compatibility: Since AutoCAD LT and AutoCAD software share the same native **`.DWG`** file format, users can exchange drawings inside or outside an organization without losing any drawing data.

In a related announcement today, Autodesk also introduced AutoSketch Release 2 for Windows, an affordable, 2D CAD-based precision drawing package (see separate news release dated 11/2/93).

To meet the varying needs of users in many industries, Autodesk now offers a family of interoperable CAD products, including AutoSketch Release 2, Generic CADD 6.1, AutoCAD LT and AutoCAD Release 12, that range in price and functionality.

Pricing, Availability and Distribution

AutoCAD LT software will be available in early fourth quarter of this year, through Autodesk Resellers and the retail channel. The suggested retail price is US$495. Autodesk is also offering a US$129 promotional transfer price for Generic CADD customers who want to move to a Windows-based product (certain restrictions may apply).

Pricing and availability for international (including Canada) and educational products vary. For the name of the Dealer nearest you, call Autodesk at 800-228-3601. Outside the United States and Canada, fax requests for information to 415-491-8303 (Latin America) or 415-491-8398 (Asia/Pacific).

System Requirements

AutoCAD LT is available for 386/486 and Pentium-based PCs with a math coprocessor, MS-DOS 3.31 or higher, Microsoft Windows 3.1 or higher, minimum 4MB RAM, 10MB available hard-disk space, permanent swap file of 2× to 4× RAM, Windows-compatible VGA display, mouse and system printer or plotter.

Company Information

Autodesk is the world's leading supplier of desktop computer-aided design automation software. Autodesk develops, markets and supports a family of computer-aided design automation software products, as well as professional scientific and multimedia software products for use on personal computers and workstations. A global company since its founding in 1982, Autodesk offers products that are now available in 80 countries and 18 languages. Autodesk shares are traded on NASDAQ under the symbol ACAD. For more information on Autodesk, please call 415-332-2344 or type GO ADESK on CompuServe(R).

– 30 –

Midcourse Corrections

From: Chris Bradshaw
To: Autodesk employees worldwide
Date: Thu, 11 Nov 93 09:05:02 PST
Subject: AutoCAD LT will NOT support AutoLISP LT

AutoCAD LT for Windows will *not* support AutoLISP LT in the final shipping version. In fact, AutoCAD LT will *not* support AutoLISP at all.

On November 1, 1993 we sent out a "Pre-Release Guide for Authorized Autodesk Dealers and Sales Representatives." This Guide included a pre-release version of AutoCAD LT and several sales and marketing documents describing AutoCAD LT. Included in the Features and Benefits section of the Guide was a table containing the AutoLISP commands that were supported in the pre-release version of the software provided in the guide. This subset of AutoLISP was referred to as AutoLISP LT. The decision has been made to remove AutoLISP LT functionality from the final shipping version of AutoCAD LT.

All other feature and product information in the Pre-Release Guide accurately describes the planned final shipping version of AutoCAD LT (i.e. there are no other changes planned).

On November 2, 1993 we announced AutoCAD LT for Windows to the press and stated that AutoCAD LT would ship in Q4 FY94. We are still on schedule to release AutoCAD LT in Q4 FY94.

Additional information on this change will be forthcoming. A final Reseller Launch Kit containing updated Features & Benefits, Q&A, etc. will be sent to everyone who received the Pre-Release Guide in the next several weeks.

If you have any questions, please feel free to contact me via email.

Thanks,

Chris

From: Chris Bradshaw
To: Autodesk employees worldwide
Date: Fri, 19 Nov 93 12:56:40 PST
Subject: AutoLISP not in AutoCAD LT

On November 11th, I sent an email to World notifying you that AutoCAD LT would *not* include AutoLISP support in the final shipping version. That is still the case—AutoCAD LT will *not* support AutoLISP.

Since then I have received numerous calls and emails on this topic. Many of you want to know what the official position is on this change and how is it being communicated to the press, dealers, distributors, and customers that might have been told that AutoCAD LT would contain AutoLISP LT. Hopefully, this email will help set the record straight.

The official reason AutoCAD LT will not support AutoLISP is:

The target customers for AutoCAD LT for Windows are seeking an easy to learn and use, affordable, design and drafting package that provides 100 percent file compatibility with AutoCAD. Recent market research indicates that AutoLISP functionality is not a requirement for these customers. Therefore, AutoCAD LT customization will include the following easy to use, user-oriented, capabilities: changing toolbar and toolbox icons, modifying pull-down menus, and creating command scripts.

Jim Quanci has already notified US developers about this change. A similar message will be going out to all US dealers shortly. PR will be contacting all members of the US press briefed on AutoCAD LT next week. Finally, a similar message will be posted to CompuServe.

For those of you in countries outside of the US, you need to be sure to notify everyone briefed on AutoCAD LT prior to November 11th about this change to the final shipping version of AutoCAD LT. A Reseller Kit for AutoCAD LT will sent out in mid-December to all who received a copy of the Pre-Release Guide for AutoCAD LT. AutoCAD LT is still on schedule to ship in Q4 FY94.

In the mean time, if you have any questions about AutoCAD LT, please feel free to contact me.

Chris Bradshaw
Product Marketing Manager

AutoCAD LT Questions and Answers

Q: What is AutoCAD LT for Windows?
A: AutoCAD LT for Windows is a new software product from Autodesk. It is a powerful CAD solution for the Windows 3.1 operating system and includes many features that will appeal to both experienced and first time CAD users.

Q: Who should buy AutoCAD LT for Windows?
A: Architects, contractors, designers, engineers, field professionals, managers, and design and drafting students

working alone or with others who need powerful 2D and basic 3D CAD at an affordable price should buy AutoCAD LT. Firms using AutoCAD that have designers, engineers, field professionals, and senior reviewers who are part of a design and production process but currently only work with paper, pencils, or printed drawings should buy AutoCAD LT. Also, AutoCAD users who want to work on AutoCAD drawing files (DWG) at home or in the field on a laptop should buy AutoCAD LT.

Q: When will AutoCAD LT for Windows ship?
A: AutoCAD LT for Windows will be available in mid-December 1993 in English. Other language versions of AutoCAD LT for Windows are scheduled to ship in early 1994.

Q: What are some of the features of AutoCAD LT for Windows?
A: AutoCAD LT for Windows includes powerful 2D and basic 3D drafting and designing tools based on the world renowned leader of CAD, AutoCAD. In addition, AutoCAD LT includes the following:

- Windows graphical user interface: pull down menus, dialog boxes, accelerator keys, and icon toolbar and floating toolbox.
- AutoCAD commands and user interface: similar command syntax and user interface to AutoCAD.
- 3D CAD features: 3D extrusions, 3D line creation, hide, shade, paper space, user coordinate system, multiple views (side, elevation, etc.), and the ability to view and modify all 3D drawings in DWG format.
- Perfect AutoCAD drawing file (DWG) compatibility: AutoCAD LT drawings are saved in the native AutoCAD format (DWG) and therefore can be exchanged with AutoCAD users with no data loss or errors.
- Performance: fast redraw, pan and zoom speed, important for large drawings (a display list driver is an integral part of AutoCAD LT).
- Clipboard support: copy and paste AutoCAD LT images and Windows Meta Files (WMF files) into AutoCAD LT drawings through the clipboard. Also, use the clipboard to copy AutoCAD LT drawings for pasting into other applications, and import text to be pasted into AutoCAD LT drawings.
- Customization support:
  - Command scripts: Scripts are ASCII text files containing sequences of AutoCAD LT commands that are used repeatedly.
  - Menus: AutoCAD LT's menus can be customized to suit user needs, and support simple DIESEL expressions.
  - Command aliases: Short aliases are provided for many built-in commands; these can be modified or augmented by the user.
  - Hatch patterns: Libraries of hatch patterns and dash/dot linetypes are supplied in text files that can be modified and augmented by the user. The file formats are the same as used by full AutoCAD; third-party vendors can supply additional patterns and linetypes as well.
  - Symbol libraries: Any AutoCAD LT drawing can be inserted in other drawings, so it's easy to create parts or symbols for re-use. Windows Metafiles (.WMF) can also be imported. Symbol libraries for various disciplines are available from third-party vendors.
  - Text fonts: AutoCAD LT supports the same text fonts (SHX files and Adobe Type-1 PostScript PFB files) as used by full AutoCAD; third-party vendors supply a wide variety of additional fonts. Toolbar and toolbox AutoCAD LT supports an optional toolbar which includes buttons to execute common commands. The commands and button icons can be customized by the user. The optional toolbox has similar customizable buttons, but its layout and screen position can be customized as well.
- Object linking and embedding: link or embed AutoCAD LT drawings into other Windows applications, like Microsoft Word for Windows or Excel.

- Aerial View: a separate, smaller Window within the AutoCAD LT graphics screen that displays a view of the entire drawing or a magnified view of a small portion of the drawing.
- On-line help: hit the F1 key at any time and receive on-line help for the current AutoCAD LT command or pull-down menu. On-line help is also now available for all pull-down menus and commands. This help is context-sensitive, with hypertext links to related topics, and is always available to the user, even in the middle of a command sequence.
- Quick Tour: under Help, the quick tour describes the AutoCAD LT user interface and common AutoCAD LT CAD operations.
- Drag and drop: drag iconized files from the File Manager onto AutoCAD LT to load them automatically. AutoCAD LT will load or execute the following file types when they are dragged and dropped onto the AutoCAD LT graphics window: DWG (insert block), DXF (load drawing), LIN (load line type), SCR (run script), SLD (display slide), and TXT (load text).

Q: What are a few of the benefits of AutoCAD LT for Windows to the *new* user?
A:

- Familiar user interface: AutoCAD LT's use of the Windows Graphical User Interface (GUI) allows new CAD users to apply their experience and knowledge of other Windows applications to learning AutoCAD LT quickly.
- Abandon the drafting board and enjoy the competitive advantages of CAD: create and update drawings quickly and accurately, reuse old designs over and over, insert common symbols in seconds, and transfer drawings electronically across the room or across the world.
- Access to customers demanding drawings in DWG file format: AutoCAD LT saves drawings in the industry standard AutoCAD DWG file format without any translation or conversion and, therefore, without any data loss. Only Autodesk can deliver a product that fulfills this promise completely.
- Comfort, security, and a broad range of compatible design automation solutions: Autodesk is the world's leading supplier of design automation software with products developed to meet the varying needs of all design professionals. If a user needs more power than is available in AutoCAD LT, they can move up to AutoCAD Release 12 and use the same data and interface immediately.

Q: What are the benefits of AutoCAD LT for Windows to an EXISTING AutoCAD user?
A:

- Work at home: AutoCAD LT for Windows only requires 4MB RAM, a perfect fit for many home PCs. The perfect DWG file compatibility ensures there is no data loss when AutoCAD drawings are edited in AutoCAD LT and vice-versa.
- Work with DWG drawings in the field: take AutoCAD drawings to a work site on a laptop running AutoCAD LT and make updates or new drawings on the spot.

Q: What are the benefits of AutoCAD LT for Windows to firms or organizations currently using AutoCAD?
A:

- Training: AutoCAD LT's ease of use and learning features make it a great training package for non-CAD users moving to AutoCAD or AutoCAD LT.
- Extend the power of AutoCAD: bring non-CAD, pencil-based engineers, designers, and managers into the development process by giving them the ability to create and modify drawings prepared by AutoCAD users.

- View/redline/plot: for users who only need to view, redline, or plot AutoCAD drawings, AutoCAD LT is an affordable solution.

Q: What are the major differences between AutoCAD Release 12 for Windows and AutoCAD LT for Windows?
A: AutoCAD LT for Windows is designed to be an out-of-the-box solution for users new to CAD who want an easy to learn yet powerful CAD program. AutoCAD Release 12 for Windows is a comprehensive design and drafting engine. R12 for Windows has many features not available in AutoCAD LT. The following is a partial list of features that are included in R12 Windows and NOT available in AutoCAD LT:

- High end drawing and editing tools (e.g. advanced geometry creation options)
- Full 3D drawing and editing tools
- AutoCAD Development System (ADS) support
- AutoLISP programming language
- Dialog box programming
- Digitizer support (mole mode)
- Plotter and accelerated display driver support (No ADI in AutoCAD LT)
- Extended entity data support
- Region modeling
- Multiple sessions
- Geometry calculator
- Dynamic Data Exchange (DDE)
- Advanced Modeling Extension (AME) support
- AutoCAD SQL Extension (ASE)
- Open Database Connectivity (ODBC) support
- AutoCAD Visualization Extension (AVE)
- Render Window and Multiple Document Interface (MDI)
- Raster and PostScript Import

Q: Why should an existing Generic CADD DOS user switch to the Windows operating system?
A: AutoCAD LT for Windows provides important productivity enhancing tools not available in Generic CADD.

For example, object linking and embedding (OLE) allows AutoCAD LT for Windows drawings to be linked seamlessly with other Windows-based application programs. Compound documents comprised of AutoCAD LT drawings, spreadsheets, and text can all be linked together to ensure the latest revisions from all applications are included in a customer report. Clipboard support (the ability to copy and paste geometry) and the Aerial Viewer are also not available in Generic CADD.

In addition, the AutoCAD LT for Windows graphical user interface (GUI) provides commonly used commands at a single mouse click, and the use of pull-down menus and dialog boxes helps new users in an organization get up to speed easily. Because mastery of one Windows-based application provides a solid foundation for quickly learning other Windows-based applications, non-CAD users familiar with Windows will be able to learn AutoCAD LT quickly.

Moving up to AutoCAD Release 12 is easy once AutoCAD LT has been mastered. AutoCAD LT commands are similar to AutoCAD Release 12 commands.

Q: How much does AutoCAD LT for Windows cost?
A: The suggested new unit price for AutoCAD LT for Windows is SRP US$495. The introductory price for

a platform transfer from Generic CADD to AutoCAD LT is SRP US$129. The introductory price for a transfer from AutoSketch to AutoCAD LT is SRP US$199. Prices in other countries will vary.

Q: Where can Generic CADD and AutoSketch users purchase a transfer to AutoCAD LT for Windows?
A: In the U.S. and Canada, transfers from Generic CADD and AutoSketch to AutoCAD LT are only available direct from Autodesk. Customers interested in such transfers should call 1-800-435-7771. Transfer policies and procedures in other countries will vary.

Q: Is there an upgrade policy for customers who wish to move from AutoCAD LT for Windows to AutoCAD Release 12?
A: There is no upgrade policy in the U.S. and Canada at this time. For information on whether there are upgrade policies in other countries, please contact your dealer or distributor.

Q: Will there be a Generic CADD for Windows product in the future?
A: No. AutoCAD LT for Windows and AutoSketch 2.0 for Windows are Autodesk's only planned Windows-based CAD products under SRP US$500.

Q: How does Generic CADD fit into Autodesk's product family?
A: Generic CADD 6.1 provides 2D drafting tools and the ability to read and write AutoCAD DWG drawing files. Autodesk will continue to sell and support Generic CADD for customers who prefer a DOS or Macintosh-based affordable CAD solution.

Q: Will there be a DOS or Macintosh-based version of AutoCAD LT in the future?
A: There are currently no plans to develop a DOS or Macintosh-based version of AutoCAD LT. However, if there is sufficient customer demand, a DOS or Macintosh-based version of AutoCAD LT may be developed in the future.

Q: Will Autodesk provide direct customer product support for AutoCAD LT for Windows?
A: In the U.S. and Canada, AutoCAD LT for Windows users will receive 90 days of free telephone support. After the first 90 days, customers will be able to purchase direct support from Autodesk. Support policies in other countries will vary.

Q: What are the system requirements for AutoCAD LT for Windows?
A:

- IBM 386/486 or 100
- An 80287, 80387 or 80487 (required with an 80486SX) math coprocessor
- Microsoft Windows 3.1 (enhanced mode)
- MS-DOS 3.31 or higher (MS-DOS 5.0 Recommended)
- 4MB RAM (minimum requirement)
- 8MB free hard disk space (2 MB additional required to install all symbol files)
- Permanent swap file 2 to 4 times RAM
- Windows-supported VGA or higher
- Windows pointing device (e.g., mouse)
- Windows system printer or plotter

Q: Is AutoCAD LT for Windows compatible with Microsoft's Windows for Workgroups and Windows NT?
A: Yes. However, AutoCAD LT for Windows does not support any Windows for Workgroups or Windows NT unique functionality. Microsoft claims that all Windows 3.1 applications will run as is with Windows

for Workgroups and Windows NT, and our testing has confirmed that AutoCAD LT is compatible with these environments.

Q: Is AutoCAD LT for Windows compatible with OS/2 2.1?
A: Yes. However, AutoCAD LT will only run in a Windows 3.1 compatibility window under OS/2 2.1. IBM claims that all Windows 3.1 applications will run under OS/2 2.1, and our testing has confirmed that AutoCAD LT is compatible with this environment.

Q: Can AutoCAD LT for Windows be run on a network?
A: Yes. AutoCAD LT for Windows will run on all networks that support and run Windows 3.1.

Q: Is a multiple-copy network license available for AutoCAD LT for Windows?
A: No. Only single-use copies of AutoCAD LT for Windows are available.

Q: Can AutoCAD LT for Windows be run in multiple, simultaneous sessions on one PC?
A: No. AutoCAD LT for Windows does not support multiple sessions.

Q: Are AutoCAD LT for Windows drawings compatible with AutoCAD Release 12 386, Release 12 for Windows, Release 11 386, and Release 11 AutoCAD Extension for Windows (AEW)?
A: Yes. The AutoCAD LT for Windows drawing format is identical to the format for AutoCAD Release 12 386, Release 12 for Windows, Release 11, and AEW, and AutoCAD Release 12 and Release 11 on any other platform.

Q: What happens to complex 3D drawings created in AutoCAD when they are brought into and back out of AutoCAD LT for Windows?
A: There is no data loss. The drawings are displayed perfectly in AutoCAD LT and all extended entity data is retained. AutoCAD LT users can zoom in and out, change the view, or perform limited editing on the 3D drawings (e.g., attach grips and stretch, add 3D polylines, dimension, snap and trim, or perform a shade). AutoCAD LT can not create new complex 3D entities (e.g., no surfaces or solids).

Q: Can extended entity data be created, attached to an entity, modified, or viewed in AutoCAD LT for Windows?
A: No. AutoCAD LT for Windows can not access extended entity information at all. However, drawings created in AutoCAD with extended entity information can be imported and exported from AutoCAD LT without the loss of any extended entity data (i.e. AutoCAD LT retains extended entity information when the file is read in and writes it back to the file when saved, if it is present in a drawing file).

Q: Will custom menus designed for AutoCAD Release 12 or Release 11 run with AutoCAD LT for Windows?
A: Yes, with the following limitation. The AutoCAD LT for Windows' menu file must be named ACLT.MNU (for short menu) or ACLT2.MNU (for long menu). Existing menus may be renamed ACLT(2).MNU. However, existing menus may contain commands not available in AutoCAD LT and those commands will not run. The default ACLT(2).MNU file can also be modified and AutoCAD LT will automatically load and recompile the menus each time it is started up.

Q: Will existing AutoLISP routines run under AutoCAD LT for Windows?
A: No. AutoCAD LT does not support AutoLISP.

Q: Will existing AutoCAD Development System (ADS) applications run under AutoCAD LT for Windows?
A: No. AutoCAD LT does not support ADS.

Q: Is there a third party developer support program for AutoCAD LT for Windows?

A: No. AutoCAD LT's customization capabilities are limited and intended to let users customize AutoCAD LT to meet their needs. AutoCAD LT is not designed to support full featured, integrated applications that have been developed for AutoCAD using AutoLISP or ADS.

Q: Which device drivers are included in the box with AutoCAD LT for Windows?
A: AutoCAD LT for Windows supports Windows-supported video display of VGA or greater resolution (at least 640-by-480 pixels), output to the Windows system printer or plotter, and input from Windows pointing devices (e.g., a mouse). AutoCAD LT should be compatible with all graphics cards, display drivers, printers, plotters, and point devices which are Windows 3.1 compatible. Because of the large number of these devices, we were unable to test all Windows 3.1 compatible devices with AutoCAD LT.

Q: Will AutoCAD Release 12 386, Release 12 for Windows, or R11 AEW specific device drivers (for printers, plotters, and displays) work with AutoCAD LT for Windows?
A: No. AutoCAD LT does not support the Autodesk Device Interface (ADI) that AutoCAD specific device drivers require.

Q: Does AutoCAD LT for Windows support digitizers?
A: No. AutoCAD LT does not support digitizers as absolute pointing devices (i.e. AutoCAD LT does not include "mole mode" support).

Creation/Evolution

Ever since, seeking venture capital in 1983, we wrote the first formal Autodesk "business plan," I'd been bothered by how blithely such documents predict things which I know from personal experience to be utterly incalculable. To run a business, one certainly needs forecasts, budgets, and plans, but making a spreadsheet for a product or company that doesn't yet exist showing revenue from sales to customers who have yet to be identified is just a waste of time, especially given how many uncontrollable factors can affect the outcome.

It amazes me that, at the very moment when central planning of national economies is thoroughly discredited and being abandoned in favour of market systems, the management culture in the West seems increasing gripped in a dirigiste *mindset right out of mid-1920's GOSPLAN—leaders of businesses who proclaim the market as supreme view the market as something entirely external to their carefully-planned and tightly-controlled enterprises. They would shudder in horror at the thought of allowing the messy, unpredictable forces of the market into their own operations—how chaotic!*

I use the word "evolution" a lot because I believe it's central to understanding how markets really work, how technologies emerge and mature, and how actual products are developed in the real world. In the early days of Autodesk, I didn't even try to guess which product would succeed—I knew I wouldn't have a hope of making such a prediction accurately. But I was pretty confident we could bat .200—that at least one out of five products we chose would succeed in the market. Then, and only then, would we focus our efforts upon the winner.

Think of it as evolution in action. We, as product developers, are creating new species, almost as blindly as the shuffling of genes, with the market—our customers—performing the winnowing process of selection. As in biology, there's no way to know how well something will work without trying it. Yet once it gets out there, you learn pretty quickly whether it was a good idea or just plain dumb.

Unlike Darwinian evolution, where selection is entirely decoupled from the mechanism of variation, products evolve in a Lamarckian manner. Training a dog to drive a car doesn't make its pups into better chauffeurs, but when a bunch of customers tell you "we want a little window in the upper left of the screen that shows the whole drawing with a little box for where we're zoomed into," the developers say, "Yes sir!" run off and hack the genome to squeeze the gimmick in, and hand it back. "Will you be needing anything else, now?"

That's why it's ever so important to get a product into the field early *and to have a rapid and responsive development and upgrade program. The first product in a category benefits from the feedback of customers and can quickly begin to converge toward meeting their*

requirements, often growing in directions not remotely anticipated in the original design. This is how AutoCAD developed—it is how every product which is successful in the long term develops—and it explains why large, mature products tend to be messy and complicated, because they have accreted, over the years, a large number of features, each requested by and valuable to, a set of customers. It is when the process of co-evolution of a product with its customers and the underlying technology, slows down or stops, that the product becomes vulnerable to competition. Only when a customer ceases to believe that the product he already owns will meet his needs in the future does he goes shopping for a replacement.

All this seems so obvious to me that I rarely bother trying to explain it, and yet the process by which products are proposed and developed in many organisations, including Autodesk, seems diametrically opposed to this evolutionary philosophy. Instead, we do market research (asking people what they think about something that does not exist) in order to make a detailed design, forecast market acceptance in advance, then build the product all-up to be perfect from the start.

This isn't how I learned to do engineering, and I don't think it works any better in business. The only way I know to find out what real customers want is to sell them something and then listen when they tell you what's wrong with it, and what's right with it. This process, of course, doesn't produce all the nice viewgraphs and charts and spreadsheets which are required in modern business, so it must therefore be bad.

After reviewing a number of business plans and product proposals in 1993, all of which envisioned a totally planned-from-the-start design and development process, I wrote the following piece.

Creationist Software Development

by John Walker — November 5, 1993

I think there's a "meta meta" issue that underlies the whole concept of concurrent engineering and design for manufacturing which derives from a bogus idea of how products are created which has been running rampant primarily in the U.S. for about 15 years. As far as I can tell, it is a new phenomenon, but I can't identify its intellectual roots. I know that when I try to talk about it with many people, even in conjunction with software design, I often feel like I'm explaining quantum field theory to a cat.

What I'm talking about is the "Cult of Design"—the whole bogus idea that with the proper research, our powerful intellect, marshaled by innovative management processes, and, oh yes, breakthrough design, modeling, and simulation tools, we can create, *ab initio*, products which can be mass manufactured from scratch at precisely predicted cost and quality levels, which are accepted immediately by the identified customers, and return the expected revenue to their developers.

What a pile of crap.

It took me quite a while to even appreciate the extent to which this cult of design had taken root—it really hit home when I encountered the Xerox PARC types involved with Xanadu who actually believed (and still believe today, after 8 years of failure) that they can design, in its entirety, a system which can store all the information in every form, present and future, for quadrillions of individuals over billions of years. I am not making this up—ask 'em the question directly, and that's what they'll say.

Then it slowly dawned on me that it wasn't just the Mambo Chicken[426] crowd who had hyper-warped into the techno-hubris zone—it was *everybody*, except for a few crusty old-timers like me.

I mean, when I learned engineering, the entire message was that engineering was an imprecise, underconstrained process, where there were always things you couldn't calculate and often you just had to try. Certainly, you should quantify and calculate everything you could—that's what distinguishes engineering from tinkering—but the real world is full of intractable problems that nobody has a clue how to model. Computational fluid dynamics is probably decades if not centuries away from being able to model the ignition transient in a garden-variety liquid rocket engine. So everybody in the game just figures on making and blowing up a couple dozen before they find a pattern of holes in the injector that "seems to work almost all of the time."

And when you get even a little distance from things where the physical principles are at least known, such as economics, customer preferences, or competition in the marketplace, it's a situation where there isn't a shred of a credible theory on which to base your calculations.

And yet lots of people seem to believe, for some reason, that they can predict such things.

I think it's related, in some way, to the obsession, especially in the U.S. that everything needs to be entirely risk-free and that any failure of any kind is unacceptable. Therefore, we must calculate everything in advance so that we succeed perfectly the very first time and continue to succeed precisely as predicted by our calculations.

This, I believe, is how one manages to spend US$10 billion and ten years "designing" a space station for which no hardware has yet been fabricated and for which, at the present moment, no concrete design yet exists. One can probably spend US$10 trillion on such an effort and produce nothing because the effort is as impossible as factoring a trillion digit number; one can certainly build a space station, but only by taking lots of small steps in which you make mistakes, come to understand what the real problems are and how to fix them, and build up enough confidence so that when you do finally write the check for the mondo starbase, you have a reasonable expectation that it will work pretty much as you expect.[427]

And while NASA is a master of this, you need only look at General Motors spending US$7 billion and 8 years to "reinvent the factory," and finally managing to make, in 1991, poor copies of 1985 Toyotas with engines that catch on fire.

Closer to home, folks are entirely confident that they can determine precisely what features are required in a new product to meet the needs of customers in a wide variety of industries and environments, calculate how long it will take to create such a product, determine its price point, and predict in advance how many copies will be sold and how much revenue will be returned. In fact, without a "business plan" which claims to do this, no product development effort will obtain funding.

When this process totally fails, and it always does, that doesn't seem to weaken the belief in a "design process" which, in reality, is as bogus as astrology. It's always a bad manager, problems with tools, etc.—*precisely* the unpredictable factors which make *a priori* design impossible in the first place.

Absolutely the only way I know to succeed with an innovative product is to throw something together quickly, push it out the door, persuade some lunatic early-adopters to start using it, and then rapidly evolve it on a quick turnaround cycle based on market acceptance and driven by a wish list from actual users.

[426] See "Great Mambo Chicken and the Transhuman Condition" by Ed Regis. Reading Massachusetts: Addison-Wesley, 1990. ISBN 0-201-56751-2.

[427] Of course, Autodesk's own fling in the space station business (see page 396) hasn't produced anything to date, so one mustn't be too nasty to NASA.

Every successful product I have been involved with, either as a developer or as a user, seems to have followed this path. I think that when people beat up Microsoft for shipping shoddy early releases, they just don't recognise that early releases are shoddy *inherently,* because they haven't had a chance to evolve based on customer requirements. (Of course some are shoddy due to plain old bugs that should have been found during development; that is inexcusable.)

That is certainly the development strategy Parametric has followed with Pro Engineer, and I would put the probability of beating that product with a "right the first time better solution" at exactly zero.

> Study it forever and you'll still wonder. Fly it once and you'll know.
>
> —Henry Spencer

> . . . I have not failed. I've just found 10,000 ways that won't work.
>
> —Thomas Edison

> We are the products of editing, rather than authorship.
>
> —George Wald

I've pretty much given up trying to explain this old-time engineering philosophy to people—they just don't seem to get it. So these days I just try to avoid projects that work that way and concentrate on what is disdained as "maintenance programming" or "hacking" of the kind that produced AutoCAD.

In June 1991 I started to write a screed titled, "The Evolutionary Paradigm—Engineering, Management, Markets, and Choice" which would attempt to explain all of this so clearly that it might have a chance at turning the product development culture at Autodesk into which I believed were more productive modes. I really think that this design cult is nothing more than a branch of creationism which thinks its members are so omniscient that they have no need for market-driven evolution to perfect their efforts. Evolution is messy, unpredictable, and utterly unmanageable. Its sole advantage is that it works.

After writing a few paragraphs, I gave up and went back to work on AutoCAD—I quickly realised how hopeless it was to convey these concepts when I had found myself incapable, in person, of explaining them to people I knew to be smarter than I am.

Ten Years After

Well, that's it, then. We close the book on the continuing story of Autodesk as 1993 draws to an end. Ten years before, at the end of 1983, Autodesk was wrapping up its first full year of operations, confident that in AutoCAD we had a winner on our hands. AutoCAD, which a year before had been only dimly perceived as Autodesk's primary product, had generated more than a million dollars in revenue, rendered the company profitable, appeared on the front page of leading PC magazines, and created an operating company with offices, payroll, dealers, and almost a thousand customers.

Ten years later, a million dollars in sales is hardly worth noting; Autodesk sells more than that, on average, every day of the year. AutoCAD, then so ill-defined even in the minds of the programmers who were writing it, has become the global standard for computer aided design, with more than a million users and billions of drawings created with it.

Autodesk, the company so weird that no venture capitalist could screw up the courage to fund, is now part of the Standard and Poor's 500 average, the very benchmark of U.S. industry. The company that *Business Week* called, in 1985, a "high-flyer that may not fly" was named, by that same publication, the Number One Hot Growth Company in America in 1986 and 1987.

Success is what comes at the end of a sufficiently long sequence of failures. Success is no guarantor of continued success. Often it is merely a punctuation mark in the series of small failures that are the day to day reality of building a business and creating new markets.

How perverse, yet strangely equitable reality is. One must fail repeatedly in order to succeed, but one can fail, on the grand scale, only after having first succeeded. The history of Autodesk is the history of many failures and a few great successes; so it is with any business that succeeds at all. It is the groping of fallible humans, with incomplete and often inaccurate information, toward uncertain decisions which will ultimately be judged by a market whose preferences are unknowable in advance and evolve with every product introduced. It is perverse, indeed, that success conditions us to avoid risk, when unwarranted risk-aversion leads directly to the bullseye of failure.

In reading this book you've seen Autodesk grow from the point where one sale of our product for $1000 increased the company's bank balance by more than 5% to where it's "lost in the round-off" amongst revenue of a million dollars a day. You've seen the hopes and dreams of a few individuals grow into a company which has made the likes of IBM, Computervision, and Intergraph tremble and redefine their own mission. And you've seen the problems that success can bring which, albeit unpleasant to experience, are surely preferable to the problems of failure.

You've seen individuals grow when confronted with increasing challenges, and you've seen individuals, myself among them, decide that the challenges before the company required they consign its destiny to those better

equipped to manage it. You've seen the consequences of decisions made, confronted by an unforgiving marketplace, yet constrained by Wall Street expectations totally at odds with developing a successful business amidst a technological revolution.

And yet, it didn't come out all that bad, ten years on, did it? If our naïve dreams of owning the software business didn't come true, our more realistic hopes of controlling one of the major application areas surely have. As that area matures and approaches its inevitable point of saturation, the challenge before Autodesk is to mobilise the same creativity and energy that created AutoCAD in 1982 and 1983, but on a 1990's scale, to create the products and build the markets which will define the personal computer industry in this decade and beyond.

If Autodesk rises to this challenge, the Autodesk story will not end with this chapter.

Appendix A

Before Autodesk

Because Autodesk has been successful, there's a tendency to forget what a high-risk undertaking it was to start the company. Most of the founders of Autodesk were involved in preexisting ventures of their own, some while also holding down full-time jobs. Embarking on Autodesk meant abandoning these ventures, some of which looked quite promising at the time, for a new business in an untested market.

Starting Autodesk wasn't the only opportunity that beckoned at the time. I wrote this paper in late 1981 to plot the strategy of Marinchip and the people around it, who encompassed a large percentage of the Autodesk founders. This strategy represented the "safe evolutionary path" for Marinchip and would, had it been pursued, have led to utter failure.

It's worth keeping this in mind when evaluating new business opportunities that seem to diverge from Autodesk's traditional areas of success. Special thanks are in order to David Ciari, who typed in this entire document from the only existing paper copy.

Product Development Strategy Working Paper
Revision 0—September 29, 1981
by John Walker

This paper describes the background, plans, and goals for Marinchip Systems' hardware and software development projects. This paper is being prepared as a working document for Marinchip (MS), Evolution Computing (EC), Optimistic Systems (OS), and Pacific Software Associates (PSA), to identify how the plans will impact work in progress by each group, how work will be distributed among the organisations, and how the work must come together before goals can be met.

This is a working document. Nothing in here is final, decided, or immutable. It exists only to serve as a starting point for discussion and as a base for the development of a formal plan.

Basic Development Strategy

MS, EC, and PSA are engaged in marketing a dead horse. The time for the 9900 to establish itself as a contender in the microprocessor sweepstakes has come and gone. We are faced always with a selling job that T.I. should

have done for us, not us for them. Our processor cannot compete in performance, address space, instruction set, or available software. No announced product from T.I., or any direction indicated in their product development holds out a hope for elimination of these problems.

Our selling point is our software. Our software is portable between processors. We should not consider ourselves tied to one processor or manufacturer because of a decision we made in the past.

We've discovered in converting the code to the Z8000 that conversion of even assembly language code poses no horrors. OS has developed the conversion tools for the Z8000, and we have learned how to best approach the problem. In addition, OS has made a native code MDEX, saving some work that would otherwise have to be done.

The Z8000 is not a good base for our future development because of the segmented memory addressing architecture, possible register set exhaustion when in segmented mode, and because the market perception seems to be that it is not the best product.

Based on our evaluation of the products available and the market's perception of them, the Motorola 68000 seems to be the best really available, second sourced, processor. Its instruction set and register architecture promises an easy conversion of our code. Its memory architecture imposes no limits on future system growth.

There are no currently available boards, either S-100 or Multibus, which implement the 68000 in a general fashion including memory management. Without the memory management chip, the 68000 can not be used in a secure multi-user mode. The memory management chip is only available in samples now, so it will be some time until boards are available.

The best way for us to work with the 68000 is to design a "node board" exactly like the one contemplated for the 9995. This board (in its final PC implementation) will have a 68000 CPU, 256K RAM (depopulatable to 128K), 4 or 8K PROM, 2 SIO, 1 PIO ports, possibly a 9512 math chip, and an S-100 I/O bus interface. The 68000 will talk to the S-100 bus only as an I/O device.

Given this board, all those with 9900 systems can start working with the 68000 immediately. NOS will support the 68000 as a node processor, so all existing 9900 peripherals may be retained unchanged, and may be accessed through the 68000 as well as the 9900. The 68000 will then also be a straightforward upgrade for our existing customer base.

The software for the 68000 node board will request all I/O through the host system processor. NOS will, of course, support this protocol. We can write a CP/M program which talks this protocol and allow the 68000 to be added to a CP/M system using all our 68000 software. The only restriction is that file names on the 68000 would then have to conform to the CP/M standards (and that JSYS requests not supported by CP/M could not be performed). All the compilers should run with no difficulties. There might be a reasonably large market for such a product.

Once the software is converted to the 68000, we can shop around for an S-100 or Multibus 68000 to serve as the host processor. On finding one, NOS/MT will be converted to run on it, using memory management to support multiple users. Of course, node board users will also be supported (if Multibus, we would have to make a Multibus node board). This work would gain us an all-68000 system (easier to sell and maintain), a cheaper entry level system (no 9900 required, no processor per user, better memory utilisation=less memory), and the ability to run programs of any size, not limited to 256K.

Details—The 68000 Node Board

The 68000 Node Board will be designed and prototyped by EC. MS will take over manufacturing, testing, and service after the final testing of prototypes. The anticipated specifications of the 68000 node board will be:

68000 CPU
256K Dynamic RAM with Parity (40 × 4164)
4K/8K/16K PROM (2 × [2716, 2732, 2764])
2 Async RS-232 ports (header pin connectors)
1 Parallel I/O port (header pin connector)
9512 Math processor (if it fits)
S-100 bus I/O interface

The node board will look like two I/O ports to the S-100 bus master. The base port address will be settable with a DIP switch. There will be a data port and a control port. Data written to the data port by port by the 68000 may be read by the S-100 bus master. The data port is bidirectional without restriction. The control port may be used by the S-100 bus master to:

Hardware reset the 68000
Interrupt the 68000
Set a status bit the 68000 can read
Clear interrupt
Clear status bit

The control port may be used by the 68000 to:

Interrupt the S-100 master
Set a status bit the S-100 master can read
Clear interrupt
Clear status bit

In addition the control port may be read to determine the value of the status bit. This allows either interrupt or non-interrupt synchronisation of data transfer between the host and the node board.

Details—Software Conversion Plans

MS will write a 68000 assembler and linker in SPL. This will initially run on the 9900.

MS will develop an NOS support module for the 68000 which will allow programs to run as under NOS. It will make requests to the master CPU for I/O. This program will probably be put in PROM once stable. Marinchip will develop support code in NOS to handle these requests. This will probably initially be a user program (which will fit in space below the system). If demand seems to justify it, MS will develop a host support program for CP/M.

Somebody will convert META and its runtime system to the 68000.

Somebody will use 68000 META to convert QBASIC 2.x to the 68000.

Somebody will convert WINDOW and SPELL using the new QBASIC (not trivial because of assembly routines).

Somebody will convert Osborne packages, Selector, etc., to 68000 QBASIC.

Somebody will convert EDIT, WORD, and other 9900 assembly programs to the 68000, carefully considering whether they should be redone in QBASIC or SPL.

EC will convert SPL to the 68000. Using SPL, Interact may be converted.

Background—Why not the 9995/99000?

We will be abandoning the 9995 node board project and the plans to redesign the M9900 CPU with the 9995. There are two major reasons for this:

First, the major problem we are having with all 9900 work is exhaustion of the 64K address space of the 9900. Neither the 9995 nor the 99000 solve this problem. The 99000 allows larger memory, and could be used with a segmentation scheme, but this is not a general solution and could not be used by unsophisticated users. We have to have a system where we can simply let the user buy more memory when his program won't fit. Thus, the major advantages of both the 9995 and the 99000 are higher performance, but neither of them delivers enough extra performance to compete effectively with the newer processors from Zilog and Motorola.

Second, The 9900 family is a largely unknown product since T.I. has failed to effectively promote it. The 99/4, considered T.I.'s last chance to establish recognition for the processor, is widely considered a flop. There is nothing in the 9900 family and nothing expected to be added which would cause a designer today to design in the 9900. Thus, the future for the 9900 is not bright. T.I. has been dropping product lines (bubble memories, watches) in response to poor market response, and the 9900 may go that way. It seems clear that if T.I. is to become a contender in the high-end micro market, it will not be with the 9900, so we would have to convert anyway. Remember, this isn't the first time this happened. The TI–ASC, for years the fastest computer in the world, only sold 7 units, 5 to T.I. divisions. Most people were unaware it existed. They dropped the product.

Background—Why not the Z8001?

Since OS has converted most of our software to the Z8002, it would be far less painful to go to the Z8001 CPU. In addition, off the shelf Multibus systems exist with Z8001 CPU, memory management, and all the large memory and support boards at excellent prices. The basic problem with the Z8000 is that it is not a general large address space processor. The Z8000 addressing is split up into "segments" from 0 to 127, and "displacements" from 0 to 64K.[1] You can address 8 megabytes by concatenating the segment and displacement into one address. Segmented mode programs can do this in both direct address pointers and in index register pairs. The problem comes in how indexing and autoincrementation is handled. When addresses are added in the processor, only the displacements are added, and the carry is discarded. Thus, if you are indexing through

[1] Does this sound familiar. . . ?

an array and cross a segment boundary, you wind up back at the start of the segment you were in, not at the next one.

As a result, you can simulate large addressing only by manually computing addresses in software, bypassing the index hardware. This is grossly inefficient. You can ignore the problem if the linker never places a module across a segment boundary and dynamic allocation never splits a buffer across segment boundaries. This doesn't help you if the user simply declares an array larger than 64K.

Further, it seems that most vendors who have looked at the Z8000 for our type of general purpose application have shied away from it and are now working with the 68000. We've had it with trying to push unpopular products.

Details—What about the Z8000 software?

OS has put a large amount of work into the Z8000 software project, and MS has supported this with hardware purchases and loans. We continue to feel that this is a valuable product, and that if properly marketed it can return not only its development costs, but also make a reasonable sum of money for both OS and MS. With the general slant of the industry being away from the Z8000, we have to target our marketing carefully, since we can't afford big splash advertising.

MS has contacted Central Data about putting QBASIC on their system. This would involve a Z8002 to Z8001 conversion as well, so we would have to work on their system to do this. They expect their operating system to be ready early next year so we should contact them then regarding that. Since we don't expect to buy one of their systems now, this may be a sticky matter to arrange.

I think we should send out new product announcements about the Z8000 QBASIC, emphasising that this is an OEM-tailorable product we want to put under customer's operating systems, and announce the development software the same way. Also, we should put together a catalogue, manual package, and OEM schedule for this and pass it on to Lifeboat so they can pitch it to OEM prospects.

There's nothing wrong with what we've done with the Z8000. It just doesn't look worth putting a lot more money and effort into unless we can generate some interest in it. If the Z8000 is going to bomb, we'd better be somewhere other than the target zone. If, say, Lifeboat uncovers a vast market for Z8000 software, we'd be happy to change our mind.

Details—What does this do to other projects?

This redirection will have remarkably little effect on the work in progress.

All work underway by PSA is conversion of QBASIC programs. These will be portable to the compatible QBASIC on the 68000. Obviously, assembly language routines should be avoided wherever possible. PSA's NOSMODEM will remain applicable as long as a 9900 host is retained, which will probably be for quite some time. Then they can convert it. Any development in SPL or QBASIC should be safe. The work on WORD will have to be carefully coordinated with the conversion. Maybe this is the time to rewrite WORD in QBASIC or SPL, or maybe PSA should convert WORD to the 68000 while adding the enhancements. Discussion is

needed. PSA should avoid 9900 assembly programming like the plague.

EC will drop the 9995 node board project. The 9918A board will continue and Marinchip will market it as agreed. Marinchip's QBASIC drivers for the 9918A will not be converted to the 68000. This board will be offered primarily to existing customers who have expressed an interest in it. We won't advertise it heavily. EC will do the prototyping research on the 68000, hand wire a prototype node board, layout, tapeup, and prototype the actual node board. We anticipate a yield from the prototype run to supply MS, EC, PSA, and OS with all the 68000 node boards needed for development. EC will produce final separations, solder mask, etc., for MS to begin actual manufacture. MS will pay EC a per-board royalty on the 68000 node board, and will defray development and prototyping costs. The actual manufacturing of the node boards will be done with MS existing vendors and contractors and will not consume any EC resources, except for possible revisions and corrections as needed. EC may undertake the META conversion, and will convert SPL. FORTRAN should be developed with the large address space of the 68000 in mind. EC and MS should consult over features needed in the new linker to support FORTRAN.

OS can use a node board in the Z8002 system and work on 68000 code. Depending on the market we can stir up for the Z8000 code, OS will continue to develop and support that code. The Z8000 C should be evaluated for portability to the 68000, as that would be a very desirable product to have. Optimally, we could consider converting the OS machine to a native 68000 processor (when we can get one), and letting OS do the 68000 host support (NOS/MT conversion). This would not affect work with or use of the node board. We can lend OS a 9900 to bridge the conversion gap, if needed.

MS has no development work in progress, to speak of. We'll be busy enough converting code, we reckon, without taking on anything else.

Details—Whither the 9900 and its customer base?

With the emphasis on the 68000, Marinchip will cease to actively market the 9900 system to new customers. We will continue to support existing customers and OEM's, and we will continue to sell and service all our existing 9900 products. OEM's using our system will not be impacted by this change in direction, except if they anticipate future 9900 based products, which will not be forthcoming.

We will continue to market the 9900 software through Lifeboat, and if a new 9900 OEM walks in the door, we will not turn him away. However, we will be up-front about where we are going and what this means.

Our large-system OEM's will be encouraged to configure and expand systems by adding 68000 node boards. Our converted software should make this relatively painless and advantageous. We hope to keep these OEM's with Marinchip by offering them the same software, much better performance, less big system slowdown effects, better reliability, and a migration path which doesn't make them throw away either hardware or software.

We want to make it clear, though, that if an OEM is happy with the 9900, he can continue to get them for the foreseeable future.

We will announce all of this in the *Shifting Bit*[2] in a message for end users. We will basically say that support continues, sales continue, but most new products will be 68000 based. We'll explain the reasons and show how a user can upgrade at minimal hardware cost.

[2]The *Shifting Bit* was the independently-edited newsletter for Marinchip users. Marinchip subsidised its printing and mailing.

Details—Marketing strategy

As soon as we have the software on the node board, we will begin a multi-pronged marketing strategy.

We will market the 68000 utilities (QBASIC, SPL, WINDOW, WORD, etc.) under other operating systems by directly contacting other 68000 vendors and through Lifeboat. This would be sales on an OEM buyout or on a royalty basis.

We will announce a 68000 "system" and advertise it. This system will have a 9900 running NOS in 64K as the support processor.

We will announce the 68000 node board as an add-on to our existing customers and OEM's. We will emphasise its use in expanding or building large NOS configurations.

We will advertise the 68000 node board as a CP/M addition. We will supply CP/M program and disc so our software can be used on the 68000.

What we want to do is to sell the 68000 software to OEM's, and possibly (say under XENIX) to end users. We'll try lots of alternatives to get the visibility we need to find them.

Details—Who pays and who gets what?

The current structure of the MS–OS–EC–PSA community is a complete mess. Before we all get into this stuff we should figure out exactly how everybody's going to be compensated for what they do.[3] MS intends to continue being the "shock troops" of the community. We'll do the manufacturing, testing, advertising, mailing, phone answering, shipping, and receiving.[4] OS, EC, and PSA will thus be freed to get development work done (although they are certainly welcome to do scuzz work if they wish!!). The cleanest arrangement for MS is to simply handle all work by OS–EC–PSA on a royalty basis, paid per item sold, credited upon sale. We would define the royalty as a percentage of the payment received by MS for the item sold, so discounted OEM sales would be attractive to MS to negotiate for.

MS will pay for the hardware and other out of pocket expenses involved in getting into the 68000. We will provide node boards to OS and PSA at cost, with payment deferred until offset by royalties. (EC, of course, will just keep a prototype of the node board).

MS would like to renegotiate the agreements for MIDAS software from PSA to the standard percentage royalty. This isn't to grab more money, it's to remove the ambiguity in OEM sales (e.g., if the royalty is a fixed price and we discount the package to 50% off to sell 20,000 of them, we'd end up losing money on the deal, so we can't pursue it). Also, we could promote those packages more aggressively.

Further, we must normalise the status of those packages initially developed and nominally owned by MS, which others have or are planning to enhance. We need to somehow make work on those packages pay off via a royalty or upgrade charge from sales of the package. We can't afford to let sidelines build up for these packages, especially if we're going to be moving them between processors.

[3] It was precisely the messiness of this kind of issue that persuaded me that it was essential to start a new company with everything spelled out explicitly.

[4] This idea was carried over, essentially unchanged, into the original organisation plan for Autodesk.

We can help all these goals by unbundling the 68000 software as much as possible. If we eliminate the "free software" concept, we can at least handle royalty on a reasonable basis.[5]

None of this implies that there's anything wrong with developing products and selling them directly. It's just that if we're going to go after large volume OEM sales, it's to the advantage of the developers of software to have it visible to all potential customers who may examine our software. We need to make something on the software we sell that way to defray the costs of learning it enough to answer questions and intelligently market it, to do the front line support, and to handle updates, etc.

Details—Things to be resolved

The following are things we need to pin down.

How much will it cost us to build the node board? How much should it sell for? What will be the problems making and testing it?

Who will do what in converting the software? What items are on the critical path? What other priorities are contending for the time of these people? What interest in the revenue from a piece of software will converting it earn?

How should the completed plans be best presented to dealers and OEM's? What is the best way to announce and advertise the products? What do we do about one-off user inquiries?

What exactly will be MS's role in selling Interact and PSA-developed software? Are EC and PSA prepared to provide full end user support? If not, how much will it cost MS to do so?

How can we use the conversion to the 68000 to make the next conversion easier? What should be rewritten in a higher level language now? What language?

Should we put OS's C on the 68000? Whitesmiths is supposed to have a C for the 68000 soon. Should we use it?

What about the INS16000? How about the IAPX432? Should we make node boards for them also? How real are they? Is software for them in demand? What pieces can we sell?

What is the maximum capital drawdown we can expect before we find out if this was a good idea or not? How long should it be until we know? If MS is putting up all the development money (and hence taking all the risk), what is reasonable compensation for that? What do we do if it doesn't work?

[5]Because of the nonexistence of a microcomputer software market in 1977, and the prevalence of software piracy, Marinchip had adopted a strategy, then close to unique, of bundling most of the system software with its CPU board, thus funding the development of the software through hardware sales.

Appendix B

The View from Sweden

Before the founding of Autodesk, Lars Åke Moureau was the proprietor of Smådatorinstitutet AB in Onsala, Sweden. Lars' company was one of Marinchip's very first dealers, and throughout Marinchip's history, one of the most successful. As few documents chronicling the growth of Autodesk outside the United States were available for this compilation, Lars contributed this view of Autodesk from the other side of the ocean. Unless otherwise identified, the footnotes are mine.

The notes I have from Autodesk and my participation in it are a mixture of business and personal matters. I am deliberately leaving out all personal (noninteresting) comments. The notes are written in a sort of diary form as virtually no meeting minutes were distributed. The goal is to give a viewpoint from overseas as I started to market Autodesk products in Scandinavia and later formed Autodesk AB, a wholly owned subsidiary of Autodesk, Inc.

Sweden, Fall 1981

This is it! People are now coming to *my* place to take a look at Interact, traveling 6 hours for a one hour demonstration. Never before in my computer career have I had a product that excites people like this. Interact is unusable as-is but with some extensions and new hardware it could make it.

Fortunately I only sold one Interact system to an existing Marinchip customer. He was never able to do any real work (PC-board) with it—but it sure was fun.

London, August 1982

Meeting in London with Rudolf Künzli and Richard Handyside. John Walker came later.[1] I read through a lot of material[2] for the company that later would be Autodesk Inc. John Walker brought with him his first version of MicroCAD (later renamed AutoCAD), written in PL/I (the listing could be carried in a briefcase then) and he was debugging the code as problems came up during the demonstration we had at Richard's office in London.

John Walker also brought with him some disks with a demo for a new super-machine called the Victor 9000. This was the first time I saw CAD on a "modern" PC.[3]

At the end of John Walker's stay in London, he and Richard went to a demonstration of a competing product, the Robocom BitStik.[4] It ran on an Apple II. The comment I later got was, "We drove to Dover to find the largest cliff from which to throw ourselves."[5] The BitStik was a superior product at that time. John Walker flew home and rewrote the whole code for better performance and enhancements.

Göteborg Sweden, November 1982

This was the first time AutoCAD was shown in public in Scandinavia. I borrowed a Victor/Sirius[6] computer from the Scandinavian distributor Esselte, and had a set of drawings only to run on that machine. It was a 10'×10' booth shared with an electrical utility product line. My strongest memory was when I came back to home at 4 A.M. after setting up the booth I was so exhausted I could not get out of the car. I fell asleep after pulling the handbrake.

London, January 1983

Rudolf, Richard, and I loaded Richard's Saab and drove from London to Birmingham to participate in the "Which Computer?" show. We had no booth of our own, just a desk in the Victor distributor's booth. We were now showing AutoCAD on the Victor with a touch-screen device[7] (looks like a light pen but acts nicer). We also brought a Hewlett-Packard plotter which we never got working during the show.

At this show Autodesk Europe took its first stumbling steps. Rudolf ran his business from his cellar, Richard

[1] This was the Autodesk/Marinchip dealers' meeting mentioned in Information Letter 7 (page 92), which occurred on August 7–8th, 1982. Prior to this point I had not asked Lars about joining Autodesk because in all of my conversations with him he seemed totally focused on selling hardware systems based on Texas Instruments processors. At the London meeting it was clear that he was interested in a venture like Autodesk, and we asked him to join the company.

[2] The Working Paper and Information Letters to date.

[3] Well, kinda. The Victor demo was one of the sleaziest things I have ever done. We didn't have the 8086 version even close to running on the Victor, but we wanted to be able to show the "potential of CAD on the Victor", so I wrote a plot-to-file driver for Interact on the Marinchip 9900. This wrote what we would call slide files today. Then I wrote a BASIC program on the Victor that read these slides and drew them on the screen using a Victor-supplied graphics driver for BASIC. It worked well enough to convince Victor and many Victor dealers that CAD on a PC was real—even if it wasn't real quite yet.

[4] The BitStik certainly was impressive, especially when you consider that it ran on a 64K Apple II. I brought one back from London and demonstrated it at the August 28th, 1982 meeting. Trivia lovers should note that the BitStik is also, to my knowledge, the only PC-based CAD product to ever appear in a James Bond movie (*Never Say Never Again*).

[5] Hyperbole and exaggeration. After the demo, I merely asked Richard if the Post Office tower was the best place from which to hurl my humiliated body.

[6] The Victor 9000 was sold as the Sirius outside the United States. It was the same machine.

[7] This was the SunFlex Touch Pen.

from a combined computer/bookstore, and I from a 400 square foot garage in Sweden.

At this show in Birmingham, IBM made its first official appearance with the IBM PC made in Ireland. AutoCAD was about to be ported to IBM that time, but the Victor was superior to IBM in every respect.

The Victor computer pushed AutoCAD a lot in the first years. In Sweden a company called Esselte had the distributorship for Victor and they promoted AutoCAD enormously. I believe that Esselte laid one of the cornerstones for the success of AutoCAD in Sweden.

Hannover Germany, Spring 1983

Richard, Rudolf and I participated in the Victor booth at the Hannover Fair. Chuck Peddle (the wizard of Victor Computer) came one day into the booth and he was like a movie star with the usual courtiers surrounding him.

Staying at a *gasthaus*[8] 30 miles from the show we had conversations with Autodesk in the USA at midnight. Autodesk proposed to give SunFlex (who made the touch-screen device) the exclusive right to sell AutoCAD on Victor worldwide. This was a slap in the face of all of us in Europe who were now building a dealer network. As we were all dealing directly with customers and dealers, we were developing a feeling for how AutoCAD should be sold and we knew that it was wrong to try to sell AutoCAD through hardware vendors. SunFlex got a contract excluding Europe and Autodesk probably survived at that point by large orders from SunFlex (100 systems/several times—wow!).

This has since been proven by our experience with all the big hardware manufacturers. Autodesk has had numerous OEM contracts, which helped us in the beginning as they endorsed our then-unknown product. It turned out, however, that selling CAD on a PC has its own difficulties, which require the support of a local dealer. Consequently, the OEM found himself with a big inventory of AutoCAD and decided to push hardware by discounting AutoCAD—thus upsetting the whole dealer network.

Visit to Autodesk, Inc., June 1983

This was my first visit to Autodesk Inc. Autodesk had no real office yet.[9] There was a condo with 3 IBM PCs for disc copying and a few people for distribution. We had a wishlist meeting[10] at John Walker's house and a technical meeting[11] in the South Bay.

One meeting was at SunFlex, a 20 minute trip north of the office. Nothing really came out of the meeting other than a wish list of features for AutoCAD. AutoCAD on the Victor was still outselling the IBM PC but its lead was shrinking.

[8] Richard Handyside later described the amenities of this establishment as being as close to Fawlty Towers as he ever wishes to encounter. The conversation Lars describes was conducted on the U.S. end from Mike Ford's house in sunny California, and from the European end from a phone in an unheated corridor.

[9] Hey, that *was* the real office. A month before that we were copying discs in Mike Ford's house.

[10] Duff Kurland's notes from that meeting are presented in the AutoCAD Wish List chapter, page 171.

[11] See Kern Sibbald's notes from that meeting in the June 1983 Meeting chapter, page 163.

Oslo Norway, June 1983

AutoCAD was shown by a Victor distributor. I met with a representative of a big, big company that wanted to sell AutoCAD. I began to realize what power there was in AutoCAD as a sales tool for hardware, as this company was going to force the sales of their exclusive computer by limiting sales of AutoCAD to that machine only. I also received numerous threats and intimidation from them as I tried to straighten things out. That had never happened with any computer product I carried before.

Göteborg Sweden, July 1983

I had moved to Göteborg from the rural place in the countryside, but still was a one man show. The distribution organization was beginning to come together. There were no strategy guidelines from Autodesk, Inc. about marketing, distribution, and the like,[12] so I had to create my own.

Copenhagen Denmark, August 1983

First meeting with the Danish distributor—they seemed to be very concerned and took me for a ride around Copenhagen. At this time there was no money to fly so all traveling was done by car. (I wore out two engines and got two speeding tickets during 1982–1985.)

Göteborg Sweden, December 1983

European meeting at my office in Sweden. The idea of forming wholly owned subsidiaries was discussed on an initiative from Rudolf and Richard. The outline was discussed for budgets and marketing. The premises I had were too small so I was looking for a new place. I still did not have an IBM PC, and had only 2 Victor machines. The customer base was well over 100 systems and growing.[13]

London, January 1984

Richard had his own booth at the "Which Computer?" show and it was a success. Lots of items were discussed at nights in the hotel; all meetings were informal and more of an advisory type.

Onsala Sweden, March 1984

Back in the garage again. The distributor organization was getting bigger with Norway, Denmark, Finland, Iceland, and Sweden. The workload was beginning to become unbearable.

I was still doing everything myself: translating manuals, manufacturing, accounting, additional hardware sales, marketing, literature, negotiations, demonstrations, etc. Without the help from the distributors taking care of

[12] Autonomy through neglect—long an Autodesk tradition.

[13] Lars notes that in January of 1988 Sweden sold their 10,000th copy of AutoCAD.

their territories this would have been impossible.

Basel Switzerland, May 1984

I visited Rudolf in Basel and made Swedish versions of AutoCAD 1.4. It took about 36 hours. After that I went to London and met with Al Green.[14] Al had just been hired, and coordinated the audit of the overseas offices which were going to become wholly-owned subsidiaries of Autodesk, Inc. At this time Autodesk, Inc. looked into getting capital from venture capitalists. Al was hired as Chief Financial Officer and opposed the plans of getting venture capital. The capital needed could, by better management, come from the cash flow. (Al Green, who later became president of Autodesk, Inc., managed the cash flow and later was instrumental in Autodesk's public stock offering.)

Sausalito USA, June 1984

Time for the yearly visit to Autodesk, Inc. It had now moved to bigger premises and was looking greater than ever. Sales were now beginning to ramp up and people were being hired for support, training, etc. In Europe we tried to establish some kind of policy and strategy. We still lacked guidelines from USA on how to attack the market and how to deal with OEMs and dealer networks, so we made our own policies and strategies for Europe. I was contacted by a Norwegian firm to make a port to a Scandinavian computer called Compis or Scandis. Seemed to be an easy task, but it almost got me killed. I finally got it done with help from the Swiss office.

Onsala Sweden, August 1984

Autodesk AB[15] was finally constituted. I hired 2 more people but was still in the garage. We had about 300 customers at this point.

Onsala Sweden, September 1984

Visit from the USA to Europe. Mike Ford, VP of Marketing, visited every office and it felt fine to have direct communication and make the people at the main office understand what was going on at the rural offices.[16]

[14] When Al was hired as Chief Financial Officer, he spent his first week on the job in London, trying to figure out how we could get unified finance and accounting information, which we'd surely need in the future.

[15] The Swedish subsidiary of Autodesk. Prior to the organisation of the subsidiaries in 1984 we were operating in Europe through the businesses which had earlier been Marinchip dealers, with a complicated licensing and royalty scheme. This structure grew because we had better things to do than worry about incorporating new companies. We knew, however, that if we contemplated a public stock offering, we'd have to clean up all the details, so we formally organised the European companies in 1984.

[16] When Mike Ford returned from this trip, his first to the overseas offices, he summarised it to me in the sentence, "We are very, very lucky to have those people over there".

Sausalito USA, October 1984

One distributor was trying to bypass Autodesk AB by going to Autodesk, Inc. and making a direct agreement. At the same time a Scandis computer representative wanted to have an OEM agreement directly with Autodesk, Inc. This was disturbing, and some flak was thrown. Paradoxically, this was the kickoff for Autodesk AB's growth for the future.

Onsala Sweden, November 1984

Distributor meeting at the office. Gary Wells from Autodesk, Inc. came over with new versions and all distributors were gathered. Bullet-proof vest was recommended, worn, and needed.

Kungsbacka Sweden, December 1984

Autodesk AB moved into a new location six times bigger than the garage, and 5 employees were hired. We expected this big place to fit for the next 2 years. We outgrew it in 4 months.

London, Spring 1985

Dan Drake came and demonstrated the new features of AutoCAD 2.1. He also mentioned that Autodesk were going to take over the development and marketing of AE/CADD, paying a royalty to Archsoft. As usual when you buy a product, it seemed to have less quality and more flaws at a closer look.[17]

I never dreamed at the time that I would become the product manager for the AEC product line at Autodesk, Inc., 1987–88.

Mölndal Sweden, April 1985

A company that went bankrupt left an office ready to rent. Not only an office, but I was able to take over practically new furniture and a telephone system for a fraction of their original cost. The move was done within 2 weeks and we were now in a space of 5000 square feet, employing 7 people, and were soon to hire 3 more. The distribution organization was more or less set with distributors.

Mölndal Sweden, September 1985

The first cash flow crisis appeared. Taxes, royalty, and advertising added up to a point where we had to face going to the bank and begging for a loan. After straightening up the costs everyone in the office deeply understood how easy it is to create bad, expensive habits. The lesson was painful. Luckily it came in the beginning and became a part of the culture. The ad agency was cut off in 48 hours. Actually, I have not yet

[17] And as a result, Autodesk had to essentially re-implement the package before shipping it as an Autodesk product.

found an ad agency better able to convey our "message" than we can ourselves. The best luck I have had was with an ex-technical journalist working in a PR agency.

Sausalito USA, June 1985

Public offering of the company.

Mölndal Sweden, February 1986

Mike Ford, VP of Marketing, resigned. We heard the news over the fax machine the same day. Every fast growing company has its casualties and he was one of them. The company suffered a lot but Richard Handyside from the English office went in and acted as VP of Marketing until we found a replacement.

Mölndal Sweden, March 1986

The Swedish office closed for a conference in France. It sounds crazy but despite the costs and the outrageous criticism it was worth it. In a week the employees started to meld into a company team instead of a loosely formed group of individuals.

Chicago, June 1986

The first AutoCAD EXPO—a success. Autodesk, Inc. introduced the hardware lock to the American market.

Mölndal Sweden, August 1986

I decided to take the opportunity and follow the diluted part of me that has the blood of the Vikings. Go West. From the start of 1987 I would be Product Manager for AEC Architectural and Mechanical. The Swedish office was more or less prepared and the transition went extremely well.

Sausalito, May 1988

The Autodesk story as seen from an European viewpoint. . . here is what I have learned from my experience being with Autodesk, almost from the beginning, taking the viewpoint that how we did the things we did contributed to the success we have had.

The group that started Autodesk, Inc. knew each other fairly well and shared very much the same values such as taking care of the customer.

In Europe we all had been selling hardware. (John Walker had even been manufacturing computer systems). Once we were a pure software company it was then easier to resist the temptation to start dealing with hardware

again. If Autodesk, Inc. had been dealing with expensive computers and hardware, its growth pattern would have been impossible to finance.[18]

The first years, 1982–1984, were painful in terms of mistakes, dead ends, searching for a strategy, miscommunication, etc. We maybe did not know how to do it, but we sure learned what did *not* work, and as our business was loosely defined, casual, and under our fingertips, the course was easy to correct within the minutes it took to make a phone call. In 1988 it can take a week to send out a letter—for good reasons—if the wrong message sneaks in, it is enormously difficult to correct across our huge dealer and customer base.

When business started to take off late 1984 we all had been "trained internally".

When the Viking Red Serpent was heading for Miklagard (Constantinople) to do business, he came to a fork in the road. Asking an old native the best way, the answer was, "The road you don't take is the right one".

The message is: don't mourn the "bad" decisions made in the past—they were all part of the overall deal.

[18] As so many of our competitors have learned.

Appendix C

Autodesk Trivia Quiz

Special thanks to Jamie Clay, Dan Drake, Patti Harris, Duff Kurland, Steve Mighetto, Kern Sibbald, John Sergneri, and Walt Spevak without whose encyclopedic knowledge and contributions this list would not have been possible. Answers follow on page 868.

1. What was the name of the first product Autodesk ever sold?
2. Why does the fiscal year end in January?
3. What was the original street address on Autodesk's stationery?
4. What motto appeared on Autodesk's original letterhead?
5. When and where was AutoSketch announced, and at what price?
6. What does WIDGET stand for?
7. Who drew the Nozzle, and when?
8. In what magazine did Autodesk's first full-colour ad run?
9. What was the first product launched by Autodesk *after* AutoCAD, and what obscure feature in AutoCAD was added to support it? (I'll bet 90% of even old-timers get this wrong.)
10. How much memory was required to run the IBM PC version of the first AutoCAD we shipped?
11. Roughly how many lines of source code were in the original AutoCAD?
12. Autodesk founder Mike Riddle sold his royalty rights in AutoCAD to Autodesk for $11,875,000 in January 1992, dropping his lawsuit against Autodesk. How many lines of source code did Mike Riddle contribute to AutoCAD in 1982, in return for this royalty?
13. Who bought the first copy of AutoCAD?
14. What AutoCAD command accepts the "`Q?+:$$ &9*^0E#1@2AF5+_R)!/&#<*:`" input, and what does it do? Where did "`Q?+:$$`" come from?
15. What was the name of the first dog at Autodesk? Who owned the dog?
16. When and where was the first beer-bust held, and what event did it celebrate?
17. What is the "murder at the beer-bust" story about?
18. Who was the "celebrity guest" at the party which celebrated Autodesk's first million-dollar year?
19. Who lived downstairs from Al Green in Sausalito during the Initial Public Offering?
20. Who was Autodesk's first Chief Operating Officer?
21. What did Roy Rogers want to help Autodesk do?
22. What sample drawings were used to demo AutoCAD at COMDEX 1982?
23. Where was John Walker when the doors to COMDEX 1982 opened the first day, and what was he doing there?

24. What word was misspelled in the original SOLAR[1] sample drawing?
25. Who succeeded Al Green as Chief Financial Officer?
26. What was Rube?
27. Who cracked the security of the FILETRAN machine, and how did he or she do it?
28. Who was the first national leader to see a demonstration of AutoCAD?
29. What was E-RUN ACAD, and who developed it?
30. Who kept a carbide cannon in his office, and fired it inside the building when a major project was completed.
31. Who wrote the first major review of AutoCAD, and what magazine did it appear in?
32. What public domain software package was AutoLisp based upon, and who was its author?
33. Who designed the original AutoCAD logo? Describe it?
34. Who was the founder and first editor of CADalyst?
35. When was the first AutoCAD Expo, and where was it held?
36. Which Autodesk project bore the code name "White Album"?
37. In which AutoCAD source code module(s) does Lieutenant Columbo appear?
38. Who kicked a hole in the wall of the executive office area in 2320 Marinship, and who prompted him or her to do it?
39. Who produced the first video about AutoCAD?
40. Once upon a time AutoCAD contained a driver for a plotter that worked by having a little robot car drive around on the sheet of paper carrying three pens. What was it called?
41. What was the first Unix machine to which an AutoCAD port was attempted?
42. To support which Autodesk product was the DXBIN command added to AutoCAD?
43. Which Autodesk manager fired all of his direct reports days after starting at Autodesk? Why?
44. What was the original message AutoCAD printed when an internal error caused it to crash?
45. Which Autodesk product had a check box labeled "Show trials"?
46. Who was the first Autodesk Distinguished Fellow?
47. Which AutoCAD release included an audio cassette tape?
48. What does AMIX stand for?
49. Why did the police seize a workstation containing AutoCAD source code?
50. What programming language was the first copy of AutoCAD to ship written in?
51. What is ClassWar an acronym for?
52. Which Autodesk founder, while still being paid by Autodesk, developed a competitive product for VersaCAD?
53. Why did Autodesk decide to write AutoCAD in C rather than Pascal?
54. What was FOST? What did the "F" stand for?
55. What were the REPEAT and ENDREP entities? In which AutoCAD release were they removed?
56. Which was the first release of AutoCAD to *require* a math coprocessor to run?
57. Who received the Golden Hammer award? For doing what?
58. What significant event in Autodesk's history occurred on April Fool's Day?
59. What significant event in Autodesk's history occurred on Pearl Harbour Day?
60. What was the Turbo-DigitAll project?
61. What AutoCAD-supported device worked by generating sparks?
62. When did Autodesk first establish a forum on CompuServe?
63. Who introduced the horrible "Lisp bug" in AutoCAD 2.15 which required the free update to 2.17? What was the bug?
64. Who is "Grambo?"
65. Who was the subject of a public tie-cutting ceremony?
66. Who attended the original "Marin Software Partners" exploratory meeting at John Walker's house and decided *not* to become founders of Autodesk?

[1] See page 156.

67. What C compilers have been used to produce versions of AutoCAD for the IBM PC?
68. AutoShade was originally going to use a third-party renderer. Whose?
69. On which early AutoCAD platform was the status line placed at the bottom of the graphics area? Why?
70. Which AutoCAD-supported display device did not support XORing? How were the cursor and dragging implemented?
71. What's "Electric Malcolm" and who was it named after?
72. What Autodesk product was prototyped by the Eagle Project? When was the prototype completed?
73. What device could be tested by Autodesk's QA department only after opening a window?
74. Who was the Colonel?
75. Who stuffed pizzas in his backpack at Autodesk beer-busts?
76. Why did AutoCAD jump from Version 2.6 to Release 9?
77. Which release of AutoCAD first required an 80386 or better processor?
78. Which customer reported the first bug in AutoCAD?
79. Who was the first MacArthur Fellow (future one, that is) to buy AutoCAD?
80. What was Autodesk's first product for Microsoft Windows?
81. What product resulted in Autodesk's receiving its first software patent?
82. How many ways is the word "Marin" connected with Autodesk? Cite them.
83. When was the first AutoCAD tutorial published? How was it distributed?
84. How many document-processing and desktop-publishing programs have been used to produce the AutoCAD manuals? Name them.
85. What Autodesk product was the first to sport dialog boxes and pulldown menus?
86. How many of Autodesk's founders had worked at Information Systems Design?
87. Who was the Vice President of R&D who decided, the day he was scheduled to start at Autodesk, not to after all?
88. What was the largest peripheral device ever tested by Autodesk QA?
89. Why can't AutoCAD explode differentially-scaled Blocks?
90. When was the directional sense of AutoCAD's PAN command flipped?
91. For what device was AutoCAD's AUX1 menu support developed?
92. Who had California vanity license plates "AUTOCAD," "AUTOBAD," and "ACAD OTC."
93. After what entertainment figure is the AutoCAD source module responsible for the ELLIPSE command named? How many arc segments are used to approximate an ellipse?
94. What was Autodesk's first animation product? What did it provide that the current products do not?
95. Has Autodesk had any shareware products? If so, what were they?
96. What was the "Anty Matters" project?
97. What programming language does AutoCAD's DXFIX utility support?
98. When and where was the first NAAUG conference held?
99. Who first proposed an AutoCAD Lite product? When?
100. Who wrote the memo in 1986 that led to the removal of the hardware lock.
101. What version of AutoCAD was the first to include a tablet template?
102. Which magazine described Autodesk's 1985 Initial Public Offering as a "high-flyer that may not fly."
103. Which magazine named Autodesk the "Number One Hot Growth Company in America" in both 1986 and 1987?
104. Which magazine named Autodesk as the #1 company in the world in the 1980's based on added value?
105. Where was the hot tub?
106. Which Autodesk department had an office with a fireplace?
107. What was the original U.S. list price of AutoCAD 1.0?
108. What did the original extra-cost AutoCAD option consist of? How much did it cost? How long did John Walker spend developing it?
109. How many members of Mike Ford's family worked for Autodesk?

110. Who drew the Golden Gate Bridge sample drawing?[2] What was the drawing name, and what version of AutoCAD was used to draw it?
111. Who drew the COLUMBIA sample drawing?[3]
112. What was the "carve your favorite founder" contest?
113. Which two Autodesk old-timers were named the same as pop singers. Who actually played in a band?
114. Which Autodesk product contained a rock song? Who was the artist? Name that tune.
115. As what was the Voyager spacecraft[4] mis-labeled in a magic moment in Autodesk advertising.
116. When John Walker left California for Switzerland in 1991, he sold his car and motorcycle. Who bought the car? The bike?
117. Which Autodesk officer was once an AutoCAD dealer?
118. Which Autodesk manager was previously a cabinetmaker?
119. Which Autodesk old-timer was once a ship's cook?
120. What Autodesk product was code named THUD. Why?
121. Who first designed an all-new racing car engine on a PC with AutoCAD?
122. How were the the project code names "Certain Trumpet," "Placid Impact," "Caustic Moment" etc. generated?

[2] See page 214.
[3] See page 162.
[4] See page 354.

Trivia Quiz Answers

1. AutoScreen
2. Because in a small business it lets the owners receive bonuses, reimbursements, deferred salary, etc. that they don't have to pay tax on 'till April 15 in the following year, but which the company can deduct in the current year.
3. 16 Saint Jude Road, Mill Valley, CA 94941. This was John Walker's house. Saint Jude is the patron of lost causes.
4. "Excellence in Computer Software." See page 387.
5. AutoCAD Expo—Chicago, 1986. $49.95. (AutoSketch was actually shipped at a retail price of $79.95).
6. Walker's In-Line Device Guaranteeing Elimination of Theft. The original name for the hardware lock.
7. Don Strimbu, 1983.
8. Scientific American, September 1984.
9. The AutoCAD Symbol Libraries, derived from the symbols for the AlphaMerics plotter, for which we implemented "fractional arcs" in shape definitions.
10. 128K.
11. 12,000.
12. Less than 1200.
13. Jamal Munshi, of MOMS Computing.
14. HANDLES/Destroy. "`Q?+:$$`" was the secret command one entered at Chi Corporation to initialise the FASTRAND, destroying all user files.
15. Murphy. Patti Morrison.
16. In March, 1984, at our office at 2658 Bridgeway, Sausalito, to celebrate our moving (ourselves) into that building from 150 Shoreline.
17. The janitors dumped taco sauce in a leaky poubelle sack, which made a big red spot in the middle of the floor, drips leading to the staircase and down it, and two other big pools inside and outside the front door. It was quite a surprise when I saw it next Monday.
18. Hunter S. Thompson.
19. Hunter S. Thompson.
20. Jim Pugash.
21. Go public. Roy Rogers was an investment banker at Hambrecht and Quist.
22. HOUSE,[5] SHUTTLE,[6] OFFICE,[7] and SUBDIV.[8]
23. Under the table, re-seating boards in the Marinchip mainframe, trying to get the machine to demo AutoCAD-80 working.
24. The plaque on the lunar lander on the Moon misspelled astronaut Edwin Aldrin's last name as "Alrdin."
25. Don Lohin.
26. The constraint manager based on AutoSketch that Eric Gullichsen developed.
27. Bob Elman, by discovering that **`COMMAND.COM`** on the FILETRAN machine was the same as the one on his DOS machine, then downloading **`COMMAND.COM`** from the FILETRAN machine and XORing with the copy on his boot disc, thus recovering the one-time pad.
28. Margaret Thatcher.
29. A program to defeat the hardware lock. John Burningham.
30. John Walker.
31. Davis Straub. PC World.
32. XLISP. David Betz.

[5] See page 30.
[6] See page 103.
[7] See page 61.
[8] See page 87.

33. Chuck Victory.

34. Lionel Johnston.
35. Chicago, 1986.
36. AutoCAD Release 9.
37. **`lflood.c`** and **`shading.c`**.
38. John Walker. John McLeod.
39. Jim Warren.
40. Penman.
41. AT&T 3B2.
42. CAD/Camera.
43. Al Green. Embezzlement.
44. "AutoCAD gives up." At the Anaheim A/E Systems show in 1985, VersaCAD's booth gave away buttons that said "Never gives up." We changed the message in the next release.
45. AutoShade.
46. William Bricken.
47. Version 2.5.
48. American Information Exchange.
49. The Apollo workstation in John Draper's apartment was seized when his roommate was arrested for allegedly counterfeiting BART[9] tickets.
50. PL/I.
51. Class Language Application Support System Within AutoCAD, Really!
52. Mike Riddle.
53. Because when Greg Lutz went to see a demo of IBM Pascal at the computer store, he discovered that it took 5 disc swaps or so to compile each file. Computer Innovations C let you run with the compiler in the A drive and your source in the B drive without playing disc jockey.
54. The status line displayed on 40 character graphics screens such as the CGA. "FOST" indicated Fill, Ortho, Snap, and Tablet modes.
55. They bracketed entities to be repeated in a rectangular array. They were replaced by the MINSERT command in Version 2.5.
56. Release 9.
57. Scott Heath, for implementing full 3D in Release 10.
58. Information Letter 14, April 1st, 1991.
59. First customer shipment of AutoCAD LT was announceed on December 7, 1993.
60. Chord keyboard driver, by Kelvin R. "All-thumbs" Throop.
61. The SAC Grafbar digitiser.[10]
62. May, 1986.
63. John Walker. Failing to load the stack segment and pointer registers in the proper order.
64. Greg Small.
65. Jim Stocker, on the occasion of his joining the Board of Directors. Chris Record wielded the Official Scissors.

[9] Bay Area Rapid Transit

[10] This was a digitiser in which the tip of the pen contained a small spark gap. Several times a second, a spark would be triggered which made a little "tick." The digitiser measured the time the sound arrived at two microphones and, knowing the speed of sound, calculated the position of the pen. AutoCAD supported two models, the 18 × 24 inch GP-7 and the 60 × 72 inch GP-8. A 3D model was also available. The sparks created ozone, which contributed to the electric feeling of pioneering desktop CAD. The pen was also handy for chasing away pesky cats.

66. Ernie Freeman, Tom Mahood, and John Nagle.
67. Computer Innovations, Lattice, MetaWare, Microsoft (OS/2, Windows R11 and R12), WATCOM (Windows R12).
68. Visual Engineering.
69. The NEC APC. This machine used two NEC 7220 graphics display controllers, one for text and one for graphics, the outputs of which were mixed and fed to a single screen. Screen resolution was 640×494, with the top text line devoted to a time/date display. You could turn that line off and display graphics in that area, but you couldn't display your own text (status line) there. So we placed the AutoCAD status line at the bottom of the graphics area just to gain 14 precious pixels of graphics area real estate.
70. The IBM Plasma Display. The cursor and dragging were implemented in odd vertical and horizontal pixels. REDRAW and REGEN drew to all pixels, but moving things around on the screen caused odd numbered pixels to be cleared.
71. The AutoCAD transcript facility (**XSCR** command), named after Malcolm McCullough. See page 211.
72. The Advanced Modeling Extension (AME), created by transforming AutoSolid into an AutoCAD ADS application. July 20, 1989, the twentieth anniversary of the Moon landing.
73. The Geographics digitizer. It worked using a long bar and a set of shaft encoders. When you attached it to the back of a table you needed additional space to move the puck.
74. John Kern, retired Chief Engineer of the U.S. 6th Army, who joined Autodesk as Operations Manager in 1983. Since "John" usually meant John Walker and "Kern" meant Kern Sibbald, folks referred to him as "The Colonel."
75. John Draper.
76. After AutoCAD Version 2.6, a major update which some analysts said we should have called version 3.0, everybody expected that AutoCAD 3.0 would be the release with full 3D. If we called the next release "Version 2.7" it would be perceived as a minor update and few people would notice the substantial features it offered (notably the pull-down menu and dialogue box user interface). If we called it "Version 3.0" everybody would be disappointed that it didn't contain any additional 3D functionality. So we got out of the "Version" decimal point game by just calling every update, however bright and beautiful or short and squat, a serially-numbered "Release."
77. Release 12.
78. Jamal Munshi, of MOMS Computing.
79. Wheelchair designer Ralf Hotchkiss.
80. The Autodesk Animator Player, AAWIN.
81. CAD/Camera. (The inventor was John Nagle.)
82. Dan Drake and John Walker owned Marinchip Systems, Ltd., made their investment through that company, and contributed its software as potential products. John Walker named Marinchip Systems after the last great industrial enterprise in Marin County, Steve Bechtel's World War II Liberty shipyard, "Marinship," about which he learned, in 1977, from Mike Ford. The original name for the company in the Working Paper[11] was "Marin Software Partners." Incorporated in Marin County in 1982, Autodesk subsequently moved to a waterfront building at 2320 Marinship Way, built on the site and named after the shipyard after which Marinchip was named. In 1990, Sandra Marin became Autodesk's General Counsel. In 1991, Autodesk's European Software Center opened in Switzerland, in the commune of Marin-Epagnier, canton of Neuchâtel, in an office/industrial complex named "Marin Sors." And several Autodesk founders enjoy Chinese food marinated in garlic sauce.
83. AutoCAD V1.4 was one (count 'em, 1) disk in a "Preview Pak" of ten "blank" disks in 1984. The idea was "buy the Preview Pak, run the demos, then re-use the disks." We had to cram the program and a few support and sample files onto one 360K diskette, and a tutorial into 16 of the $5\frac{1}{4}$" manual pages. At the back of the manual was a coupon for $20 off on AutoCAD.

[11] See page 13.

84. Seven: WordStar, Perfect Writer, Micro-Script, LaserScript, GML/PC, Ventura Publisher,[12] FrameMaker.
85. AutoSketch.
86. Ten: John Walker, Dan Drake, Duff Kurland, Greg Lutz, Mauri Laitinen, Keith Marcelius, David Kalish, Hal Royaltey, Jodi Lehman, Mike Ford.
87. Allen Berglund, in November 1987.
88. The Calcomp 960 plotter. It was particularly amusing to see it being tested with the first-ever laptop, the Data General One.
89. Differential scaling turns circles into ellipses, and AutoCAD, to this day, does not support true ellipse and elliptical arc entities.
90. Trick question! First on October 11th, 1982, then changed back to the original way in Version 2.0 (October 1984). See page 123.
91. The SunFlex TouchPad.
92. AUTOCAD—Greg Lutz, AUTOBAD—David Kalish (big mean black Corvette), ACAD OTC—Al Green.
93. Phineas T. Barnum. Eight.
94. AutoFlix. Worked on an EGA, worked with AutoShade, and had motion paths integrated with AutoCAD, allowed a musical sound track generated from the PC speaker, and permitted HyperCard-style button definitions for interactive control of animations.
95. AutoFlix.
96. A 1987–1988 project to develop an artificial life laboratory and related game for the Science Series.
97. FORTH.
98. Summer of 1991, San Jose.
99. John Walker, July 1984.[13]
100. Walt Spevak.
101. Version 1.0 included a blank template as a page in the manual. The plastic, ready-to-use template was introduced in Version 2.5.
102. *Business Week.*
103. *Business Week.*
104. *The Economist*, in the September 7, 1991 issue. Added value which "measures how much more a firm's output is worth than all its inputs of materials, labour and capital" was 33.9% of sales for Autodesk, and "typically 5%" for the other 2,000 publicly traded companies evaluated by the London Business School.[14]
105. Outside the "Moment's Pause" spa, adjacent to the conference room above the production office at 150 Shoreline, where weekly management meetings were held in 1984.
106. The Training Department, in the 150 Shoreline building.
107. $1000.[15]
108. Dimensioning (extremely primitive and non-associative). $500. Three days.[16]
109. Not counting in-laws, five were Autodesk employees: Mike, Joan (wife), Jerry (son), Jackie (daughter), and Stephanie (daughter). At one point there was sign in the lunch room at 2320 Marinship that said, "Your mother doesn't work here! Clean up after yourself." Some wag scrawled "Except Fords" after the first sentence.
110. Malcolm McCullough, **`SPAN.DWG`**, AutoCAD 1.3 and 1.4 (in-development).
111. Sean O'Donnell.
112. For several years, at Hallowe'en, departments competed to make, by carving a pumpkin, the best effigy of an Autodesk founder or member of management. This rapidly becane a safe way to engage in humiliation, mocking prostheses, voices, etc., and was laid to rest, to everybody's relief, after 1990's contest led to the departure of an Autodesk founder.

[12] Used for the Release 11 tutorial.
[13] See page 240.
[14] See page 690.
[15] See page 110.
[16] See page 128.

113. Al Green, Michael Jackson (was in a band).
114. The Autodesk Image Library '91 CD-ROM contained, as audio tracks, two original rock songs recorded by an Amy Wigton, *Open Eyes* and *My Own Two Feet.*
115. A turbine blade.
116. Car: Greg Small. Motorcycle: Mike Jackson.
117. John Lynch.
118. Creighton Hoke.
119. John Sergneri.
120. 3D Studio, after its principal developer, Tom Hudson.
121. Joe Shubeck of Eagle Engines.
122. By running a Unix shell script, written by John Walker for that express purpose, until something topical, evocative, or cool popped out.[17]

```
#!/bin/sh
# Create operation codenames from /usr/dict/words
#
dl=‘wc -l ¡/usr/dict/words‘
RND=‘date '+%H%S%d%M'‘
RND1=‘date '+%y%S'‘
RND=‘expr $RND + $RND1‘
bilge=‘expr $RND + $RND + $RND + $RND + $RND + $RND‘
dw1=‘expr $RND % $dl‘
dw2=‘expr $bilge % $dl‘
echo ‘sed -e ${dw1}p -e ${dw2}p -e d /usr/dict/words‘
```

[17] Yield from this script was very low, requiring tens or hundreds of runs before a suitable name was generated. Code names rejected in a typical run included, "Escherichia tether," "covariant posable," "Bellingham lint," "dragonhead pollster," "disperse nucleotide," "inclusion schism," "detonate manual," "identity provocateur," "ordain swipe," "dairyman heroin," and "hickory mighty." A few more runs produced the code name for this work, "Obstruct Shrilly."

Appendix D

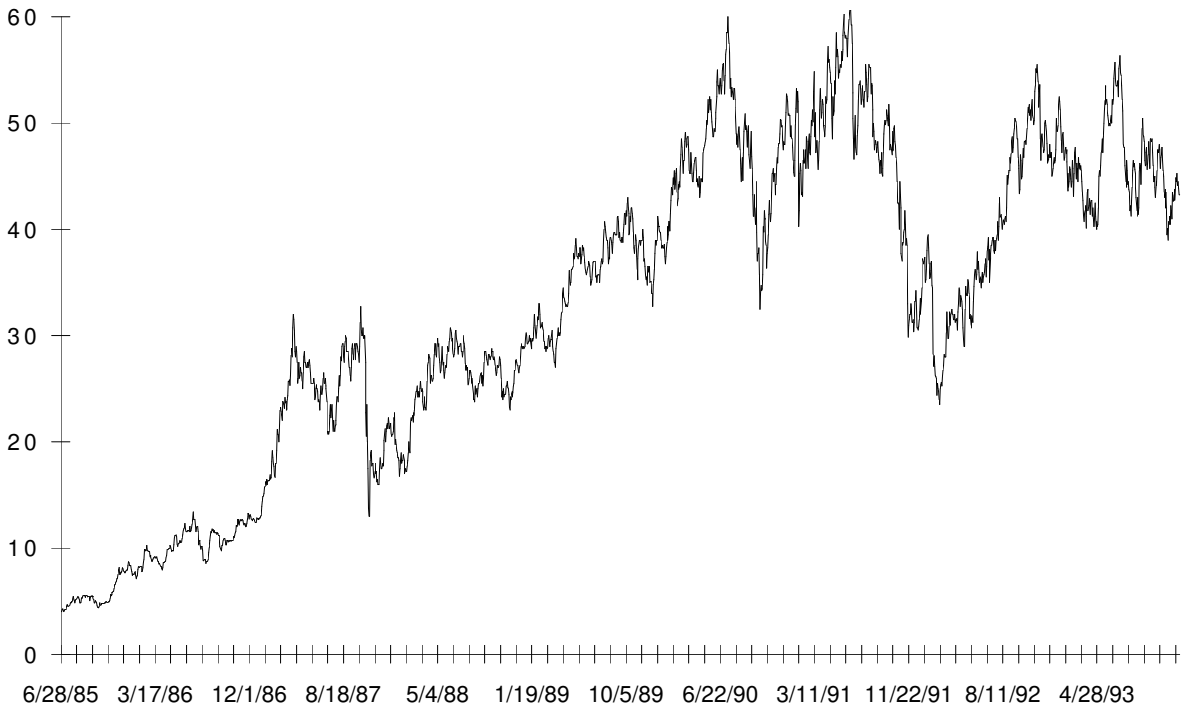
Financial Results

Qtr.	Ending	Sales	Profit or (Loss)	Earnings per Share
83Q1	1/31/83	14,733	(9,465)	
FY 83		14,733	(9,465)	
84Q1	4/30/83	98,156	23,248	
84Q2	7/31/83	176,571	24,358	
84Q3	10/31/83	320,504	20,695	
84Q4	1/31/84	593,961	44,702	
FY 84		1,189,192	113,003	.01
85Q1	4/30/84	1,262,614	222,902	.01
85Q2	7/31/84	1,971,812	337,756	.02
85Q3	10/31/84	2,307,373	342,726	.02
85Q4	1/31/85	4,332,175	723,154	.04
FY 85		9,873,974	1,626,538	.10
86Q1	4/30/85	5,066,345	1,114,149	.08
86Q2	7/31/85	6,760,507	1,437,276	.09
86Q3	10/31/85	7,709,159	1,765,902	.11
86Q4	1/31/86	9,994,928	2,202,507	.12
FY 86		29,530,939	6,519,834	.34
87Q1	4/30/86	11,026,852	2,428,295	.12
87Q2	7/31/86	12,172,459	2,713,982	.13
87Q3	10/31/86	13,651,207	3,018,388	.14
87Q4	1/31/87	15,531,284	3,460,602	.16
FY 87		52,381,802	11,621,267	.55
88Q1	4/30/87	17,292,330	3,880,423	.18
88Q2	7/31/87	18,778,230	4,815,648	.21
88Q3	10/31/87	20,208,126	5,435,855	.23
88Q4	1/31/88	22,978,410	6,409,677	.27
FY 88		79,257,096	20,541,603	.89

Quarter Ending		Sales	Profit or (Loss)	Earnings per Share
89Q1	4/30/88	25,757,000	7,246,000	.30
89Q2	7/31/88	27,253,000	7,599,000	.31
89Q3	10/31/88	30,292,000	8,506,000	.35
89Q4	1/31/89	34,000,000	9,344,000	.39
FY 89		117,302,000	32,695,000	1.35
90Q1	4/30/89	38,752,000	10,800,000	.45
90Q2	7/31/89	43,390,000	11,132,000	.46
90Q3	10/31/89	46,436,000	11,919,000	.49
90Q4	1/31/90	50,013,000	12,527,000	.51
FY 90		178,591,000	46,378,000	1.91
91Q1	4/30/90	53,552,000	13,449,000	.55
91Q2	7/31/90	56,218,000	14,523,000	.59
91Q3	10/31/90	60,841,000	15,583,000	.63
91Q4	1/31/91	59,792,000	13,200,000	.53
FY 91		230,403,000	56,755,000	2.30
92Q1	4/30/91	67,114,000	15,666,000	.63
92Q2	7/31/91	70,079,000	16,372,000	.65
92Q3	10/31/91	70,255,000	16,242,000	.65
92Q4	1/31/92	66,526,000	11,902,000	.38
FY 92		273,974,000	57,794,000	2.31
93Q1	4/30/92	75,360,000	9,313,000	.39
93Q2	7/31/92	85,479,000	10,558,000	.43
93Q3	10/31/92	93,755,000	12,808,000	.50
93Q4	1/31/93	98,560,000	11,194,000	.44
FY 93		353,154,000	43,873,000	1.76
94Q1	4/30/93	101,665,000	15,442,000	.62
94Q2	7/31/93	103,613,000	16,471,000	.65
94Q3	10/31/93	98,176,000	14,928,000	.60
94Q4	1/31/94	102,142,000	15,325,000	.63
FY 94		405,596,000	62,166,000	2.50
Total		1,731,268,000	340,075,000	

Appendix E

Autodesk Stock Price History



Pre-Public Offering Stock Valuation

Date range	Share price
4/29/82–11/8/83	$0.022222
11/9/83–8/21/84	$0.066667
8/22/84–4/14/85	$0.166667
4/15/85–6/10/85	$0.333333
6/11/85–6/27/85	$2.000000

Stock Splits

Date	Factor
July 1983	10 for 1
May 1985	3 for 2
March 1987	3 for 1

Appendix F

Product Release History

Autodesk Product Initial Shipment Dates

Product	Date Shipped	Date Terminated
AutoCAD-80	December 1982	October 1984
AutoCAD-86	January 1983	
AutoScreen	January 1983	February 1984
AE/CADD / AEC Architectural	August 1985	May 1990
CAD/camera	July 1985	December 1987
AutoSketch	October 1986	
AutoShade	September 1987	August 1993
AutoCAD AEC Mechanical	August 1987	May 1990
AutoFlix	March 1988	
AutoSolid	June 1988	October 1990
Cellular Automata Lab	June 1989	November 1993
Autodesk Animator	October 1989	February 1994
Chaos—The Software	November 1989	February 1994
Animator Clips	June 1990	
3D Studio	October 1990	
Home Series	July 1991	
Animator PRO	July 1991	
HyperChem	March 1992	January 1994
Graphic Impact	May 1992	November 1993
3D Concepts for Windows	June 1992	November 1993
Office Layout	September 1992	November 1993
Instant Artist	September 1992	November 1993
Design/Manufacturing Expert	February 1993	
Cyberspace Developer Kit	February 1993	
Aemulus CADAM Emulator	July 1993	
AutoVision	October 1993	
AutoCAD Data Extension (ADE)	October 1993	
AutoCAD LT	December 1993	
AutoCAD Designer	January 1994	

AutoCAD Version Release Dates

Release	Version name	Release date
1	Version 1.0	December 1982
2	Version 1.2	April 1983
3	Version 1.3	August 1983
4	Version 1.4	October 1983
5	Version 2.0	October 1984
6	Version 2.1	May 1985
7	Version 2.5	June 1986
8	Version 2.6	April 1987
9	Release 9	September 1987
10	Release 10	October 1988
11	Release 11	October 1990
12	Release 12	June 1992

Kelvin's Holiday Hacks[1]

Year	Code name	Description
1985–1986		AutoLisp entity access
1986–1987	Urgent Fury[2]	Pull-down menus and dialogues
1987–1988	Certain Trumpet	Curves and sculptured surfaces
1988–1989	Placid Impact	AutoCAD database recovery and audit
1989–1990	Caustic Moment[3]	Moving IGES from AutoCAD core code

[1] See page 505 for a fictionalised description of the "Holiday Hack" tradition.
[2] See page 363.
[2] See page 512.

AutoCAD Major Feature Release History

Feature	Version	Date
ADE Package	1.4	October 1983
ADE–1	2.0	October 1984
ADS (AutoCAD Development System)	11	October 1990
Apollonius problem	2.5	June 1986
Arcs, alternate	1.4	October 1983
Array command	1.4	October 1983
Attributes	2.0	October 1984
Audit	11	October 1990
AutoLisp	2.18	January 1986
AutoLisp entity access	2.5	June 1986
Axis command	1.4	October 1983
B-Splines	10	October 1988
Binary DXF	10	October 1988
Blipmode command	2.1	May 1985
Boundary hatching	12	June 1992
Break command	1.4	October 1983
Cascading menus	12	June 1992
Chamfer command	2.0	May 1985
Colour and linetype by entity	2.5	June 1986
Configuration	1.3	August 1983
Cramer's rule	12	June 1992
Cursor menu	12	June 1992
Curve fitting	2.1	May 1985
Database, portable	9	September 1987
Dialogue boxes	9	September 1987
DIESEL	12	June 1992
Digitiser calibration, nonuniform	12	June 1992
Dimensioning	1.2	April 1983
Dimensioning, associative	2.6	April 1987
Dimensioning, extended	2.0	October 1984
Dimensioning, styles	11	October 1990
Divide command	2.5	June 1986
Doughnut command	2.5	June 1986
Dragging	2.0	October 1984
Drawing file recovery	11	October 1990
Dtext command	2.5	June 1986
DXB	2.1	May 1985
DXF	1.0	December 1982
DXF, new format	2.0	October 1984
Ellipse command	2.5	June 1986

Continued...

Feature	Version	Date
Explode command	2.5	June 1986
Extend command	2.5	June 1986
Extended entity data (Xdata)	11	October 1990
External (shell) commands	2.1	May 1986
External references (XREF)	11	October 1990
Files command	1.4	October 1983
Fillet command	1.4	October 1983
Filmroll command	2.6	April 1987
Geometry calculator	12	June 1992
Grid, rotated, isometric	2.0	October 1984
Grips	12	June 1992
Handles	10	October 1988
Hardware lock, international	2.1	May 1985
Hardware lock, domestic	2.5	June 1986
Hatching	1.4	October 1983
Hide command	2.1	May 1985
Initial release	1.0	December 1982
Kanji font support	2.5	June 1986
Layer names	2.0	October 1984
Line types	2.0	October 1984
Main menu eliminated	12	June 1992
Measure command	2.5	June 1986
Menus, icon	9	September 1987
Menus, pull-down	9	September 1987
Mirror command	2.0	October 1984
Network file locking	11	October 1990
Object snap	2.0	October 1984
Offset command	2.5	June 1986
Pan reversed	1.0	December 1982
Pan reversed again	2.0	October 1984
Paper space	11	October 1990
Periodic drawing save	12	June 1992
Personalisation	11	October 1990
Plot preview	12	June 1992
Plot to file	2.5	June 1986
Point acquisition filters	2.6	April 1987
Points, fancy	2.5	June 1986
Polyface meshes	11	October 1990
Polygon command	2.5	June 1986

Continued...

Feature	Version	Date
Polylines	2.1	May 1985
PostScript import/export	12	June 1992
Printer plotters	2.1	May 1985
Programmable dialogues (Proteus)	12	June 1992
Programmable status line	12	June 1992
Qtext command	2.0	October 1984
Raster file import/export	12	June 1992
Region modeler	12	June 1992
Rendering, built-in	12	June 1992
Repeat/Endrep removed	2.5	June 1986
Rotate command	2.5	June 1986
Scale command	2.5	June 1986
Scripts	1.4	October 1983
Selection, interactive	2.1	May 1985
Shade command	11	October 1990
Shape compiler	2.0	October 1984
Sketch command	1.4	October 1983
Slides	2.0	October 1984
Slide libraries	9	September 1987
Solid modeling (AME)	11	October 1990
Spline curves	9	September 1987
SQL	12	June 1992
Stretch command	2.5	June 1986
Style command	2.0	October 1984
Surface mesh entities	10	October 1988
Tablet configuration	2.0	October 1984
Text alignment, general	11	October 1990
Text fonts, multiple	2.0	October 1984
3D Level 1	2.1	May 1985
3D Level 2	2.6	April 1987
3D, full	10	October 1988
Time command	2.5	June 1986
Transparent commands	2.6	April 1987
Trim command	2.5	June 1986
Undo and Redo	2.5	June 1986
Undefine/Redefine commands	9	September 1987
Units command	1.4	October 1983
User coordinate systems	10	October 1988
Variables and Expressions	2.1	May 1985
Viewports, multiple	10	October 1988
View command	2.0	October 1984
Zoom and pan, fast	2.5	June 1986

Index

3D Studio, 725, 790, 872

Action at a distance, 492, 494
Added value, 690
ADS, 512
Advanced Drafting Extensions (ADE), 128
Advanced Modeling Extension (AME), 355, 725, 767, 870
Advertisements
  super programmers, 305
Advertising
  agencies, 219, 254
  copy, 219, 252, 254, 305, 392
AE/CADD, 264, 292, 861
Airheads, 256, 336, 385, 656, 801
Aldrin, Edwin E., Jr., 868
Altamira Software Corporation, 355
Alternative minimum tax, 246, 275
AME, 355, 725, 767, 870
American Information Exchange, Inc., 497, 726, 745, 795, 810, 869
AMIX, 497, 726, 745, 795, 810, 865
Analog, 412
Anderson, Poul, 244
Animator Pro, 726
Annual week of rest, 363, 382, 505, 718
Ansermet, Ernest, 710
Anty Matters, 871
Apollo 17, 252
Apple II, 86, 857
Archsoft, 435, 861
Aristophanes, 513
Arrogance, 530
Artificial life, 491, 575, 871
Aurora 1000, 147
Auto Book, 92, 97
AutoCAD, 71, 89, 104, 110, 260, 263
  3D, 207, 382, 389, 431, 455
  acquisition, 67
  applications, 266, 298
  BASIC "demo", 857
  bug, first, 122
  constraint manager, 727
  database, 372
  demo version, 137
  development log, 114–137
  first bug, 866
  first sale, 864
  for Windows, 724, 756, 781–783
  "gives up", 869
  limits, 116
  Lisp, 250
  Lite, 240, 871
  logo, 865
  LT, 832, 869
  Macintosh, 433
  Oops command, 117
  Pan command, 116
  Release 10, 382, 389, 423, 431, 453, 455
  Release 11, 434, 781
  Release 12, 45, 726
  Release 9, 363, 389
  Russian edition, 453
  screen menu, 116, 124
  Snap command, 116
  source code, 263, 298
  source distribution, 398
  source licensing, 93, 596
  Swedish version, 860
  tablet calibration, 117
  tablet menu, 116, 866
  transcript feature, 211
  user interface, 302, 363, 437
  version 1.4, 860
  version 2.1, 861
  wish list, 171
AutoCAD AEC, 435
AutoCAD AEC Architectural, 264, 861
AutoCAD Expo
  Africa, 582

Moscow, 453, 582
Prague, 582
Autodesk
directors, 65
fellows, 419, 865
fiscal year, 864
global village, 703
incorporation, 64
naming, 47, 203
officers, 64, 290
reorganisation 1991, 659–683, 750
trivia, 864
Autodesk (product), 41, 59, 64, 68, 71, 85, 89, 109, 153, 169
Autodesk AB (Sweden), 856, 860
AutoFlix, 871
AutoLisp, 250, 263, 293, 834, 835
Automata, cellular, 491–496, 727, 768
AutoScreen, 40, 59, 72, 90, 111, 153, 868
AutoShade, 310, 436, 869
AutoShield, 330
AutoSketch, 240, 243, 327, 436, 864, 868
for Windows, 726, 757
AutoSolid, 355, 423, 453, 502, 870
AutoSurf, 813
AutoThing, 341
Aver, Carolyn, 733

Bacteriophages
F2, 577
T4, 576
Balinson, Frank, 707
Barnes, Bill, 453
Barnett, Peter, 11, 239, 259, 273, 279, 295
Barnum, Phineas T., 871
Bartz, Carol, 807, 811, 812, 814
Beans, 219
Beer bust, 868
Benchmark, Autodesk, 40, 324, 560
Bernoulli box, 398
Bethea, Carl, 340
Betz, David, 868
Bézier patches, 318
Bicycles, 338
Binnig, Gerd, 569
Birge, Robert, 573
Boeing 777, 799
Bond, James, 857
Boulton, Sandra, 255
Bozos, 219, 254, 336
Bradshaw, Chris, 834, 835
Bricken, William, 869
Browne, Harry, 18
Buffett, Jimmy, 274
Buffett, Warren, 498
Bugs, *see* Fixes
Bush, Barbara, 725
Bush, George, 719
Business plans
John Walker's, 189
Kern Sibbald's, 193
prospectus, 260
Business Week, 185, 417, 732, 871
Business, shape of, 468

C, 23, 91, 261, 865, 869
CA 84 conference, 494
CAD
computer science, 307
CAD/camera, 265
CAD/Camera, 256, 869
CAD/camera, 292
Cadence, 692
Cadetron, Inc., 355
Cake, piece of, 219
Carbide cannon, 372, 865
CBASIC, 23
Cellular automata, 328, 491–496, 727, 768
Chambers, Frank, 233
Chaos—The Software, 695, 745
Chevrolet, 336
Chi Corporation, 868
Chicago Board of Trade, 473
Chicken, great mambo, 796, 844
Churchill, Winston S., 353
Ciarcia, Steve, 145
Ciari, David, 11, 848
Clarke, Arthur C., 429, 791
ClassWar, 865
Clough, Eric, 240
COBOL, 649, 798
Columbo, Lieutenant, 865
Commodity Trading Package, 73
Communicator, 73, 85
CompuServe, 865
Connolly, Ruth, 707, 733

Constraint management, 436
Conway, John Horton, 492
Corporate image, 653
Cosmic perspective, 394
CP/M, 23
Creationism, 842
Cringely, Robert X., 560
Cromwell, Oliver, 351
Crosby, Philip B., 143
Cult of design, 842
Cuneo, Richard, 501, 707
Cyberspace, 424, 439–452, 726, 748, 786

Davies, Malcolm, 340, 489, 692–694, 733
Davis, Scott, 707
dBASE II, 41, 73
Dea, Larry, 243
Dealer channel, 587, 595
Dealers, 302
Debt, 62, 277, 405, 478, 585
deGrado, William, 572
Design, cult of, 842
Desktop video, 788
Dible, Donald M., 20
Diff, 72
Digital Research, 188, 203
Dimensioning, 128, 147, 871
Diversification, 287, 396, 694
Documentation, 85, 104
Drake, Dan, 49, 57, 86, 90, 106, 139, 145, 149, 157, 189, 193, 244, 277, 335, 337, 343, 420, 424, 505, 861
Drake, Stillman, 424, 505
Draper, John, 869
Drexler, K. Eric, 558, 570
Dyson, Esther, 726

E. coli, 577
Eagle project, 767, 866
Ears, 254
Economics, quantum, 469
Economist, The, 690, 871
Edison, Thomas, 845
Efficient markets, 472
Egg gesso, 789
Einstein, Albert, 657
Eisenhart, Mary, 733, 784
Electric Malcolm, 211
Electronic mail, 91, 112
Elkis, Dick, 168, 696
Elman, Bob, 295, 374, 868
ENIAC, 170, 440, 565
Equity, 277
Equlibrium, 472
Esselte, 858
Ethernet, 158
European Software Centre, 45, 709, 726, 870
Evolution, 842
External Tanks Corp., 396

Farmer, Randy, 500
FASTRAND, 868
Fawlty Towers, 858
Fear, 277
Feynman, Richard, 492, 568, 657
FILETRAN, 81
Final Days, The, 600
Fixes, *see* Bugs
Ford, Mike, 80, 90, 93, 145, 152, 189, 202, 222, 290, 778, 779, 860, 862
Fort, Charles, 468
Foundyller, Charles, 386
Fredkin, Edward, 495
Freeman, Ernie, 870
Fulbright, Cary, 832
Fulton, Bob, 489

Galt, John, 372
Gates, Bill, 597, 601, 695, 705, 823, 824, 826, 827
Generic CADD, 386, 489, 725
Generic Software, Inc., 489, 694
Gibson, William, 446, 449
Glasnost, 408
Goering, Hermann, 276
Golden age of engineering, 392, 406, 650, 714, 729, 783, 809
Golden Gate Bridge, 214
Golden hammer, 455, 865
Goldklang, Lew, 11, 330, 340, 345, 707
Goldman, Lisa, 733
Goldmann, Peter, 44, 85
Gordon, Bill, 582
Graphic Development Labs, 147
Gravitational assist, 231
Greed, 277
Green, Al, 340, 472, 489, 529, 691, 721, 733, 860, 871, 872

becomes chairman, 420
becomes president, 349
Gregory, Roger, 416
Guerrilla programming, 22
Gullichsen, Eric, 451, 868

Hacker's Diet, The, 795
Hallowe'en, 871
Halobacterium halobium, 573
Hand, invisible, 462
Handyside, Richard, 85, 120, 123, 137, 166, 290, 453, 857
Hannover Fair, 858
Hardware lock
blue sky, 165
debating, 345
original name, 868
removing, 359
return of, 524–529
Heath, Scott, 382, 431, 455, 869
Hoke, Creighton, 872
Holiday hacks, 363, 505, 878
Home series, 725, 753
Humboldt, Alexander von, 493
Humour, 47, 270, 271, 319–342
HyperChem, 534, 558, 725, 782, 791, 793, 798
Hypercube, Inc., 534, 558
Hypertext, 416

IBM, 224, 793
IBM PC/AT, 297, 373
IGES, 157, 512, 803
Illustrations
Adder, 138
AutoCAD logo, original, 869
Blivet, 42
Columbia, 162
Earth globe, 415
ENIAC ring counter, 170
Expense report, 218
Flag raising, 717
Flowchart, 56
Function plot, 427
Geological strata, 239
Golden Gate Bridge, 214
Highway, 259
House, 30
Isomech, 273
Mountain bike, 243
Nozzle, 206, 864
Office (2D), 61
Office (3D), 210
Open plan, 310
Pump, 279
Schematic, 388
Selexol, 79
Separated at birth?, 780
Shuttle (2D), 103
Solar system, 156
Staircase, 295
Subdivision, 87
Teapot, 318
Voyager, 354
Image, corporate, 653
Industrial revolution, 564
InfoCorp, 296
Information letter
1, 31
2, 43
4, 57
5, 64
6, 80
7, 88
8, 104
9 (apocryphal), 221
10, 221
11, 227
12, 280
13, 49, 349
14, 600, 647, 648, 659, 795
Insight Automation Ltd., 58
Intel 8086, 83
Interact, 856
Intergraph, 325
International Herald Tribune, 828
Investments, 274, 276
Invisible hand, 462
IPO, 260
Isaacson, Portia, 726

Jackson, Mike, 872
Jadrnicek, Rik, 202
Jingoism, 755
Jones, K.C., 691
Jude, Saint, 868
Junior Woodchuck Manual, 691

Kafka, Franz, 719
Kahn, Phillippe, 695, 705
Kalish, David, 89, 95, 147, 158, 871
Kapor, Mitch, 421, 726, 746, 804
Kern, John, 167, 189, 203, 204, 222, 778
Kleinn, Volker, 340, 707, 709, 718, 728, 739, 762
Kowal, Charles, 424, 505
Kropotkin, Peter, 221
Künzli, Rudolf, 44, 45, 166, 857
Kurland, Duff, 11, 47, 90, 171, 204

Laitinen, Mauri, 90, 143
Lancaster, Don, 38
Leary, Timothy, 321, 348, 493
LeBrun, Marc, 707
Lehman, Jodi, 106
Lehrer, Tom, 649
Lenin, V.I., 280
Leverage, 277
  debt, 464
  inherent, 465
  technological, 465, 579, 690
  versus debt, 464
Liability, 33
Life, artificial, 491, 575, 871
Lifeboat Associates, 15, 84
Life, 165
Light pen, 123
Lisp, 250, 263
Localisation, 117, 177, 219, 261, 262, 299, 860
Lohin, Don, 868
Lowe, Valerie, 11
Lutz, Greg, 45, 89, 123, 204, 869, 871
Lynch, John, 691, 707, 783, 832, 872
Lyons, Eric, 326, 329, 355, 419

Machiavelli, Niccolo, 351
Maher, Kathleen, 692
Mahood, Tom, 707, 870
Mandelbrot set, 491
Marcelius, Kathy, 204
Marcelius, Keith, 43, 68, 89, 123, 165, 204
Marin Software Partners, 13, 31, 870
Marin, Sandra, 733, 870
Marinchip Systems, Ltd., 13, 84, 92, 324, 695, 848–855, 870
Marketing, 93
  strategy, 268, 269
Markets, efficient, 472
Markoff, John, 828
Marvin, Kathleen, 334
"Max Q", 580
McCall, Steve, 11
McCullough, Malcolm, 211
McDuck, Scrooge, 705
McElhaney, Ron, 524, 538
McLeod, John, 869
Meadlock, James, 325
Mechanix Illustrated, 412
Meetings
  August 1982, 106
  December 1982, 139, 145
  January 1983, 145
  February 1983, 149
  March 1983, 157
  June 1983, 163
  July 1983, 201
  October 1983, 215
  July 1985, 280
  February 1986, 290
  March 1987, 382
  April 1987, 385
  March 1988, 412
  Annual, 1988, 422
  November 1988, 455
  December 1988, 457
  September 1989, 501
  February 1990, 530
  March 1990, 531
  February 1992, 718, 765
  April 1992, 807
  new CEO, 807
  tenth anniversary, 718, 765
MES, 355, 813
Micro Engineering Solutions, Inc., 355, 813
MicroCAD, 67, 71
Microsoft, 188, 203, 793
MicroTimes, 504, 733, 784
Modeling, 307
Modeling, molecular, 534–537
Molecular modeling, 534–537
MOMS Computing, 146, 203, 868, 870
Monaco, Tony, 779
Monty Python, 731
Morale, 282, 588, 636, 675, 684, 686
Morningstar, Chip, 500

Morrison, Patti, 868
Morrow, George, 803
Moureau, Lars, 11, 166, 416, 856–863
Mouse Systems M-1, 136
Multimedia, 695, 788
Multiprocessors, 798
Munshi, Jamal, 122, 128, 131, 146, 203, 868, 870
Murphy the dog, 868

Nagle, John, 870
Naming the company, 47, 64
Nanotechnology, 491, 536, 558–579, 729, 791
NASDAQ, 472
Nelson, Theodor Holm, 416, 419, 424, 789
Neuchâtel, 709
Neutrinos, 395
New York Times, 828
Newell, Martin, 318
Nonlinearity, 466
"Number One" brochure, 252
NURBS, 813
Nydell, Stephanie, 104, 106

O'Dell, Pete, 707
O'Donnell, Sean, 162
O'Lone, Kevin, 167
O'Shea, Jack, 202, 204
Office series, 726
Opti-Calc, 59, 72
Oracle of Delphi, 649
Orchestre de la Suisse romande, 710
Orr, Joel, 693
Ostlund, Neil, 558, 798
Other peoples' money, 224
Otrona, 165, 203
Owens, John D., 110, 203

Pacific Sun, 422
Pascal, 23, 865, 869
Patents, software, 819
Patton, George S., 635, 686
Peddle, Chuck, 858
Personal holding companies, 52
Peterson, Chris, 558
Photo CD, 799
PL/I, 857
Playgirl, 419
Pohl, Frederick, 446
Pournelle, Jerry, 287
Prime Time, 254
Prodigy, 796
Programmers, super, 305
Programming languages, 60
Public relations, 254
Pugash, Jim, 868
Pumpkins, 871

QBASIC, 73, 89
Quality, 143, 159

Rand, Len, 813
Rayles, Rob, 652
Reagan, Ronald Wilson, 529, 780, 790
Record, Chris, 406, 869
Regis, Ed, 844
Rheingold, Howard, 419
Riddle, Mike, 59, 67, 153, 169, 864, 869
Roach hammer award, 271
Roach, John, 804
Robocom BitStik, 121, 241, 857
Rogers, Roy, 868
Rohrer, Heinrich, 569
Royaltey, Hal, 83, 89
Rube, 436
Rucker, Rudy, 446, 491, 768, 800
Rule 144, 278

S. Kline Newsletter, 152
Salin, Phil, 497, 796
Schumaker, Bob, 651
Science fiction, 398, 504
Science series, 491, 800
Scientific American, 695, 868
Sculley, John, 694
Sergneri, John, 872
Shakespeare, William, 349
Shareware, 866
Sheridan, Susan, 454
Shifting Bit, 456
Sibbald, Kern, 68, 85, 89, 96, 109, 153, 163, 193, 201, 389, 709
Simulation, 656
Sleazy motels, 271
Smådatorinstitutet AB, 856
Small, Greg, 869, 872
Smileys, 647
Smith, Adam, 222
Smith, Alvy Ray, 321, 355, 788

Snow, C. P., 337
Software control policy, 159
Solid modeling, 355, 813
SOLUTION 3000, 813
Sony, 204
Space station, 396, 844
Speeches
  Annual meeting, 1988, 422
  InfoCorp, 296, 437
  Neuchâtel, 709
  Windows press conference, 781
Spencer, Henry, 845
Spenser, Ken, 813
Spevak, Walt., 871
Star Trek, 323, 352, 456, 511
Stiegler, Marc, 646–652, 656, 659, 683, 810
Stock
  distribution, 62
  dividends, 477–484
  insider rules, 248
  options, 52, 187, 245, 406
  public offering, 247, 260
    road show, 271
  warrants, 62, 65, 245
Stock market
  crash of 1987, 404, 472
  crash of 1989, 502
  risks, 580
Stocker, Jim, 696, 869
Strategy, 262, 296, 408, 459, 502, 538–557, 580–599
Straub, Davis, 868
Strimbu, Don, 206, 868
Strobe plotter, 137
Stuppin, Jack, 53, 65, 73, 82, 146, 147, 158, 164, 233, 778
Summagraphics Bit Pad, 134
Sump pump, 604
SunFlex, 145, 169, 186, 857, 858
Super programmers, 305
Supernova 1987A, 395
Sutherland, Ivan, 447
Sweden, 856
Switzerland, 709

Takeover, hostile, 460, 720
Tamalpais, Mt., 258
Taxation, 66, 244, 275, 395, 478
Technology Forum, 321, 558
Terminating products, 35
Terminator 2, 804
Thatcher, Margaret, 868
Thermodynamics, 470
Thompson, Hunter S., 225, 227, 868
Throop, Kelvin R., 270, 319, 327, 328, 338, 398, 658, 869
Timbrook, Dana, 811
Time, 419
Toilets, 267
Trivia, 864
Trotsky, Lev Davidovich, 486
Tufts, Robert, 57, 778
TV Guide, 188

Ulam, Stanislaw, 495
Univac, 799
Univac 1107, 297
Urgent fury, 363
User interface, 302, 311, 437, 439–452
USI Optomouse, 134

Vader, Darth, 489
Venture capital, 221, 233
VersaCAD, 869
Victor 9000, 110, 145, 165, 219, 790, 857
Victory, Chuck, 165, 869
Video, desktop, 788
Vikings, 863
Virtual reality, 424, 439–452, 726, 748, 786
Vision, 650, 796
Visual Engineering, 436, 870
Voelz, Joel, 807
von Neumann, John, 495

Wald, George, 845
Walker, John, 13, 31, 59, 64, 80, 90, 104, 193, 221, 258, 334, 336, 363, 372, 494, 504, 512, 524, 534, 648, 653, 656, 683, 691, 700, 705, 709, 718, 733, 781, 784, 819, 842, 857, 869
  leaves chairmanship, 420
  leaves presidency, 349
  returns to programming, 343
Walker, Roxie, 158
Wall Street Journal, The, 730–780
Ware, Randolph, 397
Warren, Jim, 869

Week of rest, 363, 382, 505, 718
Weizenbaum, Joseph, 230, 445
WELL, The, 795
Wells, Gary, 310, 861
Wenig, Robert, 340
West Coast Computer Faire, 46, 59, 158
Wheeler, John Archibald, 470
Wigton, Amy, 872
Wild talents, 468, 803
Wilson, Jeff, 341
Windows, 595, 647, 723, 781, 797, 800, 833
Windows NT, 597, 797, 824
Wolfram, Stephen, 493
WYSIWYG, 311

Xanadu, 416–418, 424, 505, 792, 810, 843

Yeltsin, Boris, 719

Zachary, G. Pascal (Greg), 600, 730–780
Zhering, Brad, 649
Zlobotsky, Victor, RAdm USN (Ret.), 345