RECORD OF SOCIETY OF ACTUARIES
1993 VOL. 19 NO. 1B

TRENDS IN TECHNOLOGY

Moderator: ALLEN J. ROTHMAN
Panelists: BRIAN S. POLLACK
JOHN SARDELIS
ARNOLD F. SHAPIRO
Recorder: ALLEN J. ROTHMAN

New technological trends are emerging every day, but after an extended period of confusion due to the multiplicity of competing directions, some current trends have been brought into focus. To that extent, development using these technologies can be undertaken with confidence. The three panelists relate recent personal experiences where emerging trends in hardware, software and methodologies have been clarified.

MR. ALLEN J. ROTHMAN: The role of the actuary is changing. What used to be a back office job is now much more of a direction type of job. In some of our jobs we've taken over responsibility for determining the technology direction in our companies.

Our three panelists will be discussing different ways of looking at technology. Our first panelist, John Sardelis, of AGS Systems, will be discussing where we've been and how we got to where we are. Brian Pollack, of Milliman & Robertson, will be discussing using mainstream technology to accomplish mainstream goals. Arnold Shapiro of Pennsylvania State University will be discussing multimedia, which is a new, emerging technology, and how it can best be incorporated in our daily work.

MR. JOHN SARDELIS: I've titled my presentation, "Great Expectations." The question we'll be addressing is, Have we lost our way on the way to the information age?

The agenda will have four basic components. The first is about the perils of forecasting. The body of the presentation is about computers and productivity. I'll go through some of the research that's been done, and what lessons we can learn from it. I'll then go on to talk about information technology for the 1990s and close off with some brief comments about how this will impact the insurance industry.

Unlike President Ford, I'm a little concerned about forecasting. He seemed to have no real trepidation about it, but there's a book that was published and reviewed in *The Actuary* called *MegaMistakes*, and it talked about some of the perils of forecasting.

One example is a survey that was done in 1969 by the Industrial Research Commission. It polled research directors for their predictions for the next 10 years. It seems rather unbelievable now that it's 1993 that a group of experts could have actually predicted this. In the process of compiling some of the information, I've gone around and spoken to some actuaries and mentioned this fact to them. Some of them believe that their competition actually used these forecasts, but I can assure you, that's not really the case.

Do we have an exceptionally bad forecast here or is it the norm? I'll admit it is a fairly bad one, but the record from *MegaMistakes* suggests that, on average, we're

not really doing too well. Between 20-25% of our forecasts are accurate, 60% are mistaken, and the rest are just too vague to even quantify. The second point is that experts are no more accurate than nonexperts. This motivated me to go around and talk to the chief information officers. They're really responsible for emerging trends and technologies. I'd already established what the experts thought about the area, but I went around again and tried to get some information from the people who are really the practitioners in this field.

And finally, there is a bias towards optimism. I don't think two months go by where *Business Week* doesn't herald the next great technology whether it refers to neural nets, client/server architecture or object-oriented programming. We are always led to believe that the next great technology is going to solve our business problems. If only that were so, but I think as you'll see, that's really not the case.

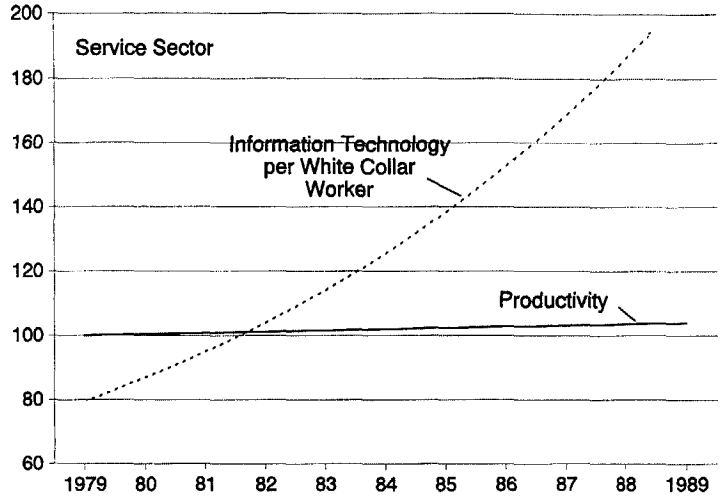
MegaMistakes goes on to say that computers are a special case. On the hardware side, I would agree with that, but that's really not the problem. The problem is software. While we've been able to reduce the cost of hardware 30% annually, and it seems likely we'll be able to do that into the distant future, it's the software that is still the sort of factor that's limiting our ability to achieve the objectives we really need to achieve.

Through the rest of my presentation, I'm going to try to stick to the three principles that I've been able to distill out of *MegaMistakes*. One is to avoid technological wonder – that old "Silver Bullet" syndrome. The Lone Ranger didn't have as many silver bullets as *Business Week* seems to have. Ask fundamental questions about markets. I don't think there's anything more critical than this. What need is satisfied? By asking this question over and over again, I think we can keep our projects and technology direction on track. Finally, make technology abide by the business rules that everyone else has to abide by and stress cost benefit analysis.

Heeding this advice, I suggest that the need that's most often noted for computers is the need to increase productivity. I look at a lot of feasibility studies and I can't think of one that fails to cite the increased productivity that is going to come about due to a new project. Unfortunately, most of these projects never do a postaudit to really find out whether that technology has truly brought about the increased productivity that's suggested. We'll look at some of the studies and see what has been determined.

There's an economist at Morgan Stanley named Stephen Roach. He has taken up, as one of his pet projects, the idea of trying to discover what has happened in the service sector and what is the productivity problem. He points out two facts that have to be dealt with. One is that the service sector purchased, in 1991, 88% of all information technology. That's a $100 billion investment. He went on to further point out that that's equivalent to about $10,000 of information processing hardware for each individual. What is the payoff for all this? I wouldn't have much of a speech if we couldn't talk about this. The productivity line in Chart 1 is that line that seems to run parallel to the x-axis. It doesn't seem to be moving all that much. But our investment in technology is moving up dramatically. This has been dubbed by Roach and many others as the productivity paradox.

CHART 1
Productivity Paradox
America's Technology Paradox



Index: 1982 = 100

Mr. Roach is not alone in this. There's an MIT professor named Gary Loveman who took a different data set and concluded the same thing.

His four conclusions are listed below:

1. There is no evidence of significant, positive productivity from IT.
2. The marginal dollar would best have been spent on non-IT inputs.
3. Much of the poor performance from IT can be explained by inadequate organizational structures.
4. In many cases, firms bought into evolving technologies or systems too soon or too heavily, when waiting might have been better.

Points one and two confirm what Mr. Roach was saying. He went a little further looking for the reasons. Point three is something that we have to pay attention to; it gives rise to the whole momentum towards reengineering, looking at the process before you really start to introduce technology. Point four is one that I think can't be stressed too firmly; that is, be aware of investing too soon and too heavily in emerging technologies.

Another MIT professor, Michael Dertouzas, has come to the defense of technology. He headed the MIT commission on industrial productivity gain. He felt that while the data did suggest strongly that there was no productivity, he offered two points as reasons. I don't believe they're too compelling, but he offered them anyway: (1) computers induced structural changes, and (2) early stages are for learning; later stages will produce productivity gains. I don't agree with that necessarily. One of the gentlemen I do business with in New York says that technology at times can be

used to turbocharge your bottlenecks. I think that there is an analogy here. Automating a poorly managed company can probably bring it to its knees. The contrapositive, if you will, states that automating a well-run company may be a way to bring about that competitive advantage that we all seem to strive for.

Mr. Dertouzas claims that in the first 20 years or so, technology was driven by supply side. A number of innovators from Silicon Valley and elsewhere would come up with some great innovative ideas and throw them out into the market. Technology would be pushed on clients so they can create their own business needs. He suggested that paradigm is changing to a demand side, or now that the businesses really have needs, they recognize those needs and they will be pulling in technology to address those needs.

He goes on further to suggest the primary trends that you have to be aware of that will shape the destiny of productivity in computers. He enumerates seven different points. There are two that I would like to call your attention to. First is a very critical one; he states that he sees no major advances in software. He's talking about development tools. That, I think, is a significant problem. Another point you should concentrate on is the idea that is going to be running through the rest of the presentation: we must find a way to lower the barriers to communication between people and computers. This seems to be where we get derailed all the time.

Mr. Dertouzas also argues that you can't really paint all technology with the same brush. In fact, he came up with a little scheme to quantify and differentiate the good technologies from the bad technologies. He took a classical formula for productivity, output over input, and, in the precomputer productivity case, defined the variables that you see below. He amortized out the equipment, such as the desk and the chair and whatever else. In the second case he said, let's add some computer equipment to this productivity measure and amortize that. Let's look at the breakeven case where P1 equals P2 and shake out these technologies.

P_1 = pre-computer productivity

$$P_1 = \frac{O_1}{I + aC_1}$$

O_1 = annual output
C_1 = noncomputer capital (desk)
a = amortization period for C_1
I = cost of labor

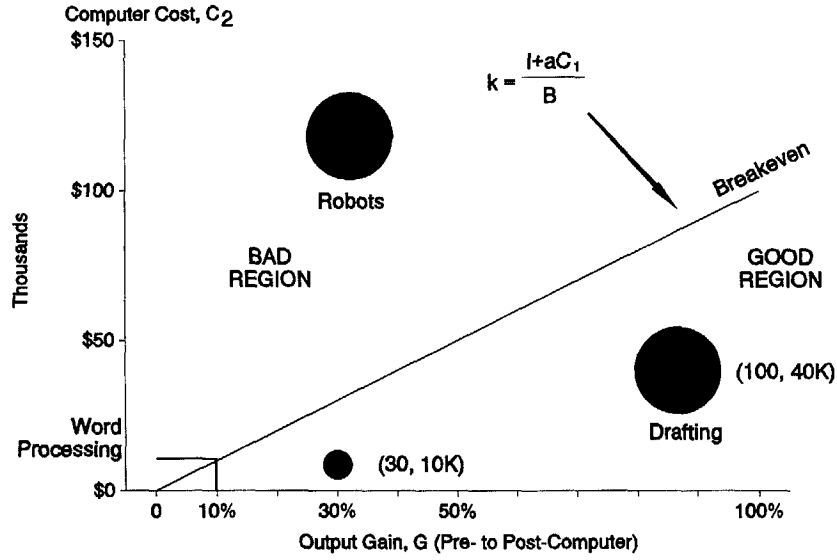
$$P_2 = \frac{O_2}{I + aC_1 + BC_2}$$

O_2 = new output level
C_2 = computer capital
B = annual computer amortization

Breakeven case: $P_1 = P_2$.

When he did this, he came up to the following conclusions: word processing is a good technology because it gives you a high enough gain to justify the expenses as does drafting, but robotics just doesn't cut it (see Chart 2). I think this is an interesting demonstration of someone actually trying to give us a way of quantifying technologies, but it's going to break down when you get to the large, multi-million dollar projects. But it imposes the discipline and that it requires us to think in terms of productivity, which is useful.

CHART 2
Pre- to Post-Computer Tradeoffs



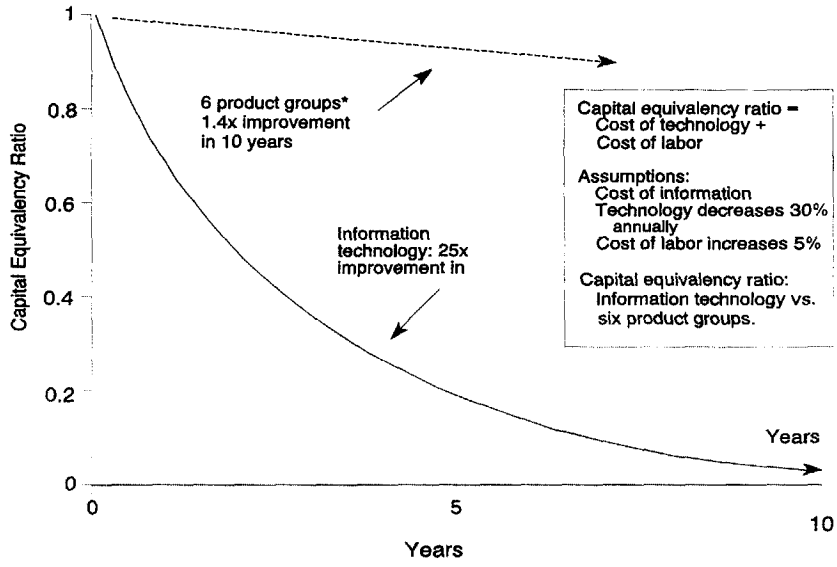
Assumptions: I = \$40,000 C_1 = \$10,000 a = .20 B = .33

We've heard a lot about productivity and the failure of computers to achieve it. I wanted to show you one last chart, and then we'll talk about some of the solutions (Chart 3). This comes from a book by another MIT professor by the name of Scott-Morton called *The Corporation of the 1990s*. The line that is sloping downward represents the cost of capital, or the cost of information technology, if you will, and it's shown a 25 times improvement in 10 years versus a 5% increase in labor and the other factors of production, which has remained generally level or sloped down.

The natural conclusion you reach from all this is, why not replace people with information technology? It seems to be a very sound approach based upon this chart. But I think we've learned from the research and the studies that I've gone to that this doesn't work. There are three lessons that can be gained from those studies. One is to avoid evolving technologies. Two is to change the organizational structure prior to installing new technology. And three is to lower the barriers of communications between people and computers. This third aspect of the puzzle, the one I think that we've missed the most, is where I would like to focus the rest of the discussion.

We're given a framework which basically says that you must consider simultaneously your strategy, your structure, your technology and your individuals or your human resource policies.

CHART 3
Capital Equivalency Ratio



*Source: U.S. Bureau of Labor Statistics, producer price index compensation per hour, nonfarm, business sector, 1950-1980.

Unless you get those four elements coordinated, you're bound to get out of phase and lose the real efficiencies that can be pumped out of automating the entire organization or creating the organization for the 1990s.

Scott-Morton has an interesting study in his book. There are six automotive assembly plants represented in Table 1. The first problem is a productivity measure. The first one is hours over units, so low is good. In the second column, we have a quality measure, which is defects per hundred units. Once again, low is good. In the third column we come to automation. This is a normalized figure. The 100 represents $650 million the GM Michigan plant invested in automation. If you go across the line, let's see what they achieved with a $650 million investment. They were the second poorest automotive assembler in terms of productivity, just a little below GM Massachusetts, and they were the worst on quality. If you look up the line, you'll see some of the people who did real well: Honda of Ohio had a mid-level automation investment but they have a terrific number for quality. If you look further down the lines, you'll see GM Massachusetts had invested very little, only 7% of $650 million or around $40 million, and got results that are a little worse in productivity, and a little better in quality than GM Michigan, which had invested $650 million. The manufacturers seem to have done something either very wrong or very right.

TABLE 1
Technology and Human Resource Management

	Productivity [a] (hrs./unit)	Quality [b] (defects/100 units)	Automation Line
Honda, OH	19.2	72.0	77.0
Nissan, TN	24.5	70.0	89.2
NUMMI, CA	19.0	69.0	62.8
Toyota, Japan	15.6	63.0	79.6
GM, MI	33.7	137.4	100.0
GM, MA	34.2	116.5	7.3

Source: Reprinted with permission from the International Motor Vehicle Program, Massachusetts Institute of Technology, John Krafcik, MIT International Motor Vehicle Program. For further discussion, Kochan, IMVP, 1988.

[a]*Productivity* herein is defined as the number of man-hours required to weld, paint, and assemble a vehicle. Have been standardized for product size, option, content, process differences, and actual work schedules (i.e. of break time).

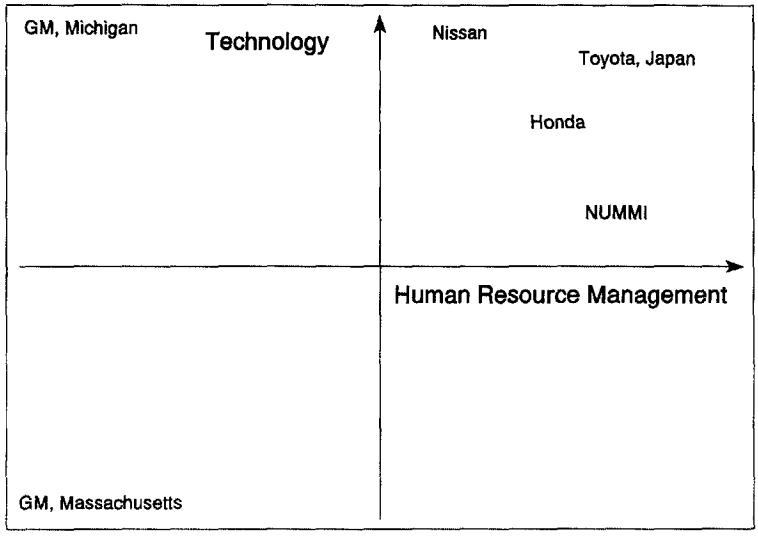
[b]*Quality* is based on a J.D. Powers survey of customized-cited defects in the first six months of ownership.

Let's see if we can determine what that was. As the research people looked at the material more closely, they noticed there was a significant variable here that seemed to cluster the groups into quadrants (Chart 4). In the high-productivity/high-quality quadrant, in the upper right-hand column, are people who spent money in human resource reform. They spent money on reorganizing into teams, education, and empowerment. GM Michigan is an outstanding case and it deserves its own quadrant. When you spend $650 million and you get the results they did, I think you belong in a box by yourself. The only other rival that they have is GM Massachusetts, who did just as poorly but spent less money achieving that objective.

Having established, at least over the last 20 or 25 years of history, what computers have done for us, I think a good question to ask is, Does anyone know this and where are the new emerging technologies headed and will they be addressing some of the problems that seem to be highlighted here? There are two groups that we're going to look at once again, heeding *MegaMistakes'* advice. We're going to look at a survey that was done of experts in the area and then we'll use my informal survey which consisted of going around Manhattan talking to some people I know and asking them what they think.

The answer is, someone seems to have been listening. The experts rank human interface technologies number one, as far as perceived organizational impact. Arnold, I'm sure, will be having more to say about that. The second element, and something we all, I think, are bombarded by daily is the *National Information Superhighway*, or the importance of communication technologies. The third item, once again, I find a little disappointing and I think it's the Achilles' heel of everything that we've gone through. We still have too much risk in major computer software development efforts. They tend to run over budget, they tend to be late, and they tend to be too costly and too risky. We still have to find a better way to develop products.

CHART 4
The Organization and Management Response



The Corporation of the 1990s

In my definitely unscientific, informal survey of the CIOs in the area, I found they seem to be more tactical, with one notable exception and this gentleman stressed usability over and over again. The CIOs (and this is both in insurance and outside of the financial area) believe that communications and system support technologies are really what they need to stress. I think the message has been lost somewhat on the CIOs.

Let's now turn finally to what this all means for insurance information trends for the 1990s. Why is information technology or computer utilization so important?

Insurance carriers, on average, utilize about 50% more information technology than other industries (Table 2). If you look at finance and insurance as a whole, they use about three times the norm. The one exception in the insurance domain are insurance agents. Their factor is 0.47 utilization. That has, I think, spawned a debate that's rattling around the insurance industry right now, whether it's really effective to give an agent more information technology. If you do, is it something that's going to be an opportunity or is it going to be a hindrance to the way they do business? There's some consideration that this is perhaps a generation gap and that, as things move forward, the agent will certainly need to understand products that are more complex and marketing techniques that are more sophisticated and will need the assistance of a computer. But that debate flows back and forth.

Heeding once again *MegaMistakes'* advice, what are the needs that information technology should be addressing? In my calls and my trips and in talking to my colleagues and people in the industry, I think I can distill it down, at least for this presentation, to three major needs (see Table 3). One is enhanced customer service, two is continued expense reduction, and three is an increased marketing focus. I

think this last one tends to be the new addition to the list. There's more and more concentration on competitive analysis, marketing thrust versus sales thrust, and the first two have been there for a while.

TABLE 2
Computer Utilization – The Potential Demand

Finance and Insurance	3.01
Banks	3.81
Credit Agencies	6.61
Securities Brokers	0.74
Insurance Carriers	1.43
Insurance Agents	0.47
Investment Holding Companies	3.94

Computer utilization is the ratio of a sector's share of the nation's high-tech capital to its share of the nation's white-collar work force.
Thus, the average utilization over the whole economy is 1.
Under-utilized sectors (where utilization is less than 0.5) are marked.

TABLE 3
The Role of Technology

	Organizational Structures	Stable and Proven Technologies	Usability
Enhanced Customer Service	Teams in a Virtual Organization	Groupware Lans	Intelligent Front-end
Continued Expense Reduction	Move to Knowledge-Worker Model	Telecommunicat-ions Workstations	Customized to Functions within Standards
Increased Marketing Focus	Separate from Sales	Consolidated Internal Database External Database	Multi-media

We should apply some of the lessons from our studies, which are, basically, to focus on the organizational structures, apply stable improvement technologies, and concentrate on one person's opinion of how this is either happening now or will evolve in the near future.

As we talk about enhanced customer service, I think more and more companies are breaking down into teams but these teams are cross-functional. A call coming into a company will be handled by a customer service representative working through a number of technologies, mostly "groupware" and local area networks (LANs), where they'll be able to answer the question using an intelligent front end. Effectively what's happening here is that the system or the front end is going to navigate through the databases. Irrespective of what line of business you're talking about, it's up to the front end to figure out to what database to go and get back to you the information you need. This will enable companies to answer on the first call, something that seems to happen very rarely in the insurance industry.

The second is the continued expense reduction. Organizationally, there's a move towards the knowledge worker model which is also part of the whole emphasis on flattening the organization. The technologies that are going to be most appropriate here are the telecommunications and the workstation technologies. What is happening is that, more and more, the functional areas are looking for functional workstations. Whether it be an actuarial workstation, an underwriting workstation or a marketing workstation, there is a certain communality that has to be engineered into it -- it should look the same, it should operate the same, it should basically be maintainable by a core group, but it should be customized to fit the particular discipline of the expert using it, be it an actuary, a marketing person or a finance person. What you'll see, I think, are workstations that have general capabilities such as electronic mail and the like, and a gateway to the regular system -- plus another component of it that has a specialized capability, a segment of it that allows a particular functionality to be enhanced through a workstation.

Finally, the increased marketing focus. I think insurance companies are starting to take the lessons from the product groups, like the Proctor and Gambles of the world, and really separate sales from marketing. They're consolidating their internal and external databases, putting them together to really understand not only the customers they have on the books, but how to basically market and integrate some of the outside information that they have. I think one of the usability factors here is multimedia. There's so much information that we have. In fact, if anything, we have too much information. What we need is a better way to manage it, understand it, project it, and deliver it. I think multimedia is one of those techniques that allows the salient features of the information to stand out.

In closing, I'd like to say that we can make the information technology pay off in a much more productive way than it has to date, but we have to keep in mind the three lessons that I think we've learned from the studies. One is to keep the person rather than the technology in mind and lower those barriers between humans and computers. Two is to do some business process reengineering before you get involved with the technology. Third, exploit existing mature technologies, put them into combinations and use that as your innovativeness. Don't get too involved with the silver bullet syndrome.

MR. BRIAN S. POLLACK: By way of background, I head up the systems and programming group at Milliman and Robertson which consists of 30 people. About half of us are systems analysts and programmers. About a quarter of us are specialists in hardware and telecommunications. The other quarter primarily have specific job roles providing specific support to our organization. I've been with the company for about nine years. I've seen a lot of changes in that time and my roots are primarily in pension systems -- pension valuations, large valuations, etc. -- although in the last couple of years I've had the opportunity to participate in a lot of insurance product development that we've undertaken.

There are probably three different kinds of people in the audience. These are different users of technology. These three groups that we have here are probably a subset of the four total I see in the industry. On the top of the heap you have creators of technology. They are a small number these days. These are the top people at places

like Intel and Novell, at Microsoft and Bell Labs. These are the people who truly do shape technology and give us the basic tools that we need to do our job.

Then come the appliers of technology. I would guess that probably a third of you are appliers. I think my group at Milliman and Robertson is appliers. These are the people that are in the industry to provide software, hardware, telecommunications support, etc. Their idea is that they can add value in a number of ways and then market their services.

A step down from them would be the users of technology. I would guess that a large number in this room are users. In your daily work probably 90% of the things that you use are purchased off the shelf and should be. I think in certain cases you find places where you need to add value, so about 10% of your work, whether you use spreadsheets or you have programming capabilities, is not designed to be reusable. It's not necessarily designed to be bulletproof, so it has a different aspect to it.

At the bottom of the heap, and I'm not sure if there's anyone who would admit to it in this room, are refugees from technology. These are the people who have decided that technology is going a little bit too fast for them. There are people better suited in their firm than they are to keep up, so they've willed it over to the next generation.

These last three groups, nevertheless, have a need to understand the general trends in the industry. The appliers have to know about the trend before it breaks so that they can be ahead of the game; the users need to know about it so that they can use packages that are efficient and not get trapped into technologies that lead them into blind alleys. The refugees from technology need an understanding of the wonders of computers these days. I do want to underscore that. Computers are a wondrous element. The more that even the refugees from technology understand about them, the more they'll be able to help their organizations use these wondrous tools.

I'm going to use a case-study approach in my presentation. I'm going to show you the odyssey of a product that had a 10-year life cycle at my firm. It's a valuation system and it has been kind of successful. Then I'm going to try to draw some trends from the life cycle of that product and some other broad trends that I see.

In order to protect the innocent I've renamed the valuation system the "theoretical valuation system," and shortened it to the TVS. Its purpose is to value pension-related plans. In terms of size, it's probably a medium or large system. It has about 300,000 lines of code. At its peak, during the development phase, it probably had six programmers working on it. During maintenance, and solely during maintenance, it probably only needs about one. During cycles when there are enhancements or extra features being added to it, they probably bulk up the group to two or three. There's also a significant number of user support people and hotline people to help support it within our firm.

I'd like to divide up the life cycle of software into four basic phases. The first is the development phase, when you're actively working to develop software. The second is an acceptance phase, where you're trying to get its use in your market or your company heightened in some way. There's a mature phase where the system peaks

out in its use. This occurs when you're comfortable with your market penetration and you're trying to support it as such. Then at the end there is, unfortunately, a declining phase where your software becomes less useful because technology has changed, or your need has changed or something significant has happened. The idea, of course, is to make the development phase as sound as possible so as to make the acceptance phase as short and brief as possible. You should extend that maturity phase as long as you can so you don't get into a case of having software that you need to throw out and rewrite.

The development of our valuation system began in the 1980s. It was the result of a need that was seen for common software because our offices were using various different software packages. We were lucky. We were able to isolate our top mathematical actuary for about a year. He was able to go off and query our consultants to determine a master set of needs. I don't want to downplay the importance of the ability to isolate someone for a certain amount of time to do this. If your development phase is sound, it will shorten your acceptance phase and increase your maturity phase to a significant degree. The factors that were driving his choices during development were simple. Computers were getting larger and faster and actuarial mathematics was recognizing the ability of computers. It was redefining itself -- it was taking fewer shortcuts, it was getting back to basic principles. Computers were allowing a more robust, larger system to come about.

Originally, we decided to house the system on a Harris supermini computer. They were used by NASA and the Navy. They were great process-control machines that were great for time sharing. At the time, it had 264K of real space, but actually only 64K was allowed for code space; the rest was for data. By today's standards, it's a small, slow machine. By the standards of that day, it was fairly fast. We chose FORTRAN as a language at that point. We chose it over COBOL because the application was going to be calculation intensive (FORTRAN is very good at that), and it didn't have as much data manipulation as would have driven us to use COBOL. Also there was a very good FORTRAN compiler available for the machine. We built it and it was accepted in our company.

We then entered the mature phase. During the mature phase you try to enhance or change or update your system in order to keep it fresh and vital in the minds of your users. We went through a number of steps to do this. The first thing we did was convert it off the old Harris machine to a Harris UNIX box. We thought we saw that micro-level machines were heading towards UNIX; because they're fast and powerful, we made the conversion. Frankly, it was a fairly painful conversion at that time. We were moving from a proprietary operating system to a totally different operating system, so we had to change all the job control. For those of you who have gone through that, you will understand the difficulties. In another sense, we were moving from one FORTRAN compiler to another. We were in the middle of the odyssey towards a new standard for FORTRAN compilers. When we started off on the first box, we were using a lot of nonstandard extensions. We learned our lesson. Over the years we've tried to get rid of everything that's nonstandard in our code, so that future ports can be made much easier.

In the mid-1980s we got on the PC bandwagon, but it didn't work out very well. We decided to move the system down to a PC and to put it within the 640K barrier.

That turned out to be impossible. We had to cripple the system so much that we had to get rid of half of it. For those of you who understand pension mathematics, we did away with all the beautiful forms of payment that we had endeavored to put in the original system, and brought it down to a lump sum payment system and annuitized everything. We did away with all the advantages we had at the beginning. It also was fairly crippling in terms of the user interface.

What did we find? Well, we found that our consultants, faced with the choice of this horrible PC system and our beautiful time-sharing system, avoided the PC system. We got smart and decided to do a real PC system and ported it down to a 16-bit system with a DOS extender that broke through the 640K barrier. This brought the full capabilities down to the PC. It was certainly slower than our mainframe version, but frankly not that much slower and it's getting a lot faster every day.

You may wonder what's going to happen with this system in the future. In a sense, we're trying to extend the maturity phase. We're going to do that in a couple of different ways. The first thing we're going to do is to get into a 32-bit version of the compiler, the question is why are we making a switch from 16 to 32 bits? Remember that this is a fairly serious application; there is a lot of movement, a lot of calculations. These days, chip technology is enabling 32-bit compilers to exist. As that happens, you'll find more and more software becoming available for the 32-bit environment.

Then you're going to find that more and more of your tools are going to be available in the 32-bit environment. The tools I'm talking about are the compilers, the debuggers, the code view products. They're all the peripheral systems that allow you to create and maintain your systems much more efficiently.

Then your operating systems, which began by supporting the 32-bit environment, start assuming that you're working in the 32-bit environment. Fairly soon you have no other choice. I think for your serious applications, you are going to have to consider that you're moving in the direction of some 32-bit operating system. I'm not totally sure what operating system that is for us. We've tested our system under OS/2 and it looks very nice. We're also a beta test site for Windows-NT. I think there's going to be quite a fight between those two systems. I have my opinions. Also, because of our history with UNIX, we're not going to turn our back on it. There are some generally stunning PC UNIX installations and more are coming out every day.

I also think to extend this maturity phase, we're going to look at a new front end. I don't agree with the word peripheral in terms of software. It's not just ornamentation in this case. User interface, as John has pointed out, really is the key to allowing your software to be used by a broader audience. It allows for ease of use by newer, less technical users. It certainly increases the speed of the training, so we're endeavoring to get a more graphical user interface. A few years ago, we chose to go with a package that's written in C and requires C programming. It's fairly object-oriented -- text and graphical. It had the advantage of being portable across DOS and UNIX platforms. For us, it was the perfect choice. In the future, the real question is, are we going to do Windows? True Microsoft Windows or NT? I have an ambivalent feeling about that. I think you do give up some portability in the process, but in the

long run I'm not so sure that's really true. I think you will find more client/server relationships popping up. You're going to find that your Windows front-end is going to front-end the guts of the program working on any number of distributed platforms.

I think you're going to gain communications if you go to Windows. For those of you who have used OLE version one, you know what I'm talking about. OLE version two is still to be released. I've seen some previews of it. It's stunning in and of itself. I think you're going to find that sooner or later there's going to be a plethora of new and good software available – things like database browsers and report writers – things that you don't want to have to write yourself. This software is going to be available for the Windows environment. They are the tools that are going to reduce your software commitments in house, or the number of things that you're going to have to worry about. Based on this, I have a funny feeling that we're going to wind up in Windows before too long.

I'm going to try to make some predictions and forecast trends. The first and most obvious trend, I think, is everything has gotten faster, smaller and cheaper. I think everybody will agree with that. No doubt about it. I think the obvious trend is the fact that we've been in a constant state of resizing our machines, of converting our programs, of porting them from one platform to another. I don't believe that's going to stop for us. There was no reason for it to stop in the past and no reason for it to stop in the future. It is and we have to accept it and I hope I have. But it is a constant process of doing such. I think a not-so-obvious trend is that these new operating systems are extremely complicated. The networking we're coming up with is extremely technical and it is not an inexpensive proposition these days to go to decentralized computing. Decentralized computing really does require decentralized expertise. You put the boxes on a person's desktop and assume it's their problem when their disk drive goes out. You have the unfortunate ramification of driving a lot of people from the user category into the technical refugee category, because they're not ready for it. You need decentralized expertise if you're going to move to decentralized computing. Your systems need to be well-written. This makes perfect sense but what does that mean? First, it needs to be well thought-out from the start and there's no substitute for this. You have to solve a need and solve it well. You must have a reason for having added value to the industry, otherwise you should have bought someone else's package.

You need good external and internal documentation in your code as well. If you have any shot to maintain your systems, you'd better have it well-documented. I would also take that to mean having some explicit debugging capabilities within your code. By debugging I mean the ability to turn on a switch and have it dump out as much of the internal calculations as possible. We've actually gone to a new system where you can set the debugging switch between one and four. If you set the switch to one, it dumps out high-level debugging statements and lets you look at them. If you can't figure it out there, you turn it to two and it puts out a second level of detail, three more, and even four more. We're experimenting with that and it's working fairly well.

I also think you need to have some kind of continuity in staff. This is a two-sided responsibility. On the one side, you must have a staff that is willing to learn and relearn. On the other hand, you're going to need a management that's willing to

accept this relearning process as a cost of doing business. We are all here for continuing education within the actuarial circles. Please understand that systems programmers and analysts have the same necessity for continuing education.

I think if you're smart you'll take small steps and try to avoid large, technical leaps. When you take a leap that is too large, you wind up getting in trouble. If you are forced to do it, I think what you should do is look for some outside help. There is a lot of outside talent these days in just about every major city near universities, these people are really top-notch professional programmers and systems analysts. These are people that do not like to work for one organization very long. They feel that it will stifle their creativity. They're not going to get to work with the foremost technology if they sign up with one firm. You can find these people fairly easily and if you're going to make any decent step in technology, I suggest you try to seek some of these people out because they can be well worth the money.

Obviously, operating systems are getting bigger on the PC level -- the 32-bit, high octane. They have security enablement; they're network conscious; and they're becoming more and more graphical. They're like the machine I started with when we started using this system, except now the cost is costing about 1/100 of what the original machine cost us.

In terms of programming, you might have heard about object-oriented approaches. I have a little different view of object-oriented programming, but first I want to draw this connection. There are two sorts of things out there that are going under the banner of object-oriented programming. One is object-oriented programming languages like C++. These are tough programs to bite into initially. They have a heavy learning curve. They're a language within a language. They're extremely powerful and they're meant for a large system – a system that is going to be maintained for a long time and for a system with authorization. On the other hand, you have packages that are truly more of an object development package. In fact, we're going to have an example of one being used here in the next presentation. These are packages that are highly object-oriented and can write the code for you behind the scenes, for example the multimedia programs. Also, if you've worked in user interface programs that have "look and feel," it will design the look and feel of your screens. Behind the scenes it will go ahead and write the code for you. We realize that these are really very different things. The object development packages are great for prototyping, for a communications package, for quick and effective use in getting something up and running as opposed to the object-oriented program languages that take a lot of time and effort to use well.

I will try to give you the last few insights. The first is to choose a strategy that's realistic for your firm, but you need to realize that there is a fairly large chasm between a technical refugee, a user, an application writer and a creator of technology. I think you need to understand and put your firm in that continuum and decide where you're going to try to be. Based on your choice you're going to have to decide what kind of people you're going to hire, the kind of career paths you're going to offer, what kind of continuing education you're going to be required to do. For your management team, decide what kind of people and skills you're going to need to manage and therefore know about. It is a requirement incumbent upon the management as well as upon the employee.

I think you need to find ways to at least monitor the essential changes that are going on in the industry. Coming to events like this is certainly important. Reading magazines is important. Seminars are important. I would even suggest that you go out and find a technology mentor in certain cases. They are available and they can do very well for your firm in some cases. I think finally, you just need to accept the fact that change is inevitable, it's continual and, I don't want to be cynical, it is wondrous these days. Computers are fairly amazing things.

MR. ARNOLD F. SHAPIRO: I'm from Penn State University. I'm going to talk about communicating actuarial ideas. I purposely limited myself to a notebook computer. It's right here in front of me. No peripherals, no big deals, no 32-bit operating system. All we're doing is communicating and I'd like to talk to you a bit about conveying actuarial concepts.

I'll talk to you a little bit about the platform that we're using. There is a continuing education program being held at this meeting at which people are getting continuing education credit by sitting in front of a computer. They're looking at touch screen monitors for people who are scared of computers. They're looking at one made by IBM, but it could be a compatible. There's a laser disk being used, a Pioneer 8000, which in half a second, can get to any one of 54,000 pictures which can be used full motion or one at a time. You can take every study that you've ever done or ever thought of doing with you on a laser disk. You can also take it with you on CD-ROM. Soon you'll be able to take it with you on just the hard disk that you carry around which go up to three gigabytes. The point is that fairly soon all this stuff is going to be very portable -- more portable than it is today. That's just a little bit by way of background. I wanted to give you a sense of what we're using. I am in a Windows environment and I am using, in this case, a software construction set. Here I'm using Toolbook, but Authorware will let you, for example, cross platforms. There are all kinds of things available. Whatever you feel comfortable with is the thing to be looking at.

You need help, of course, if you're going to do this. Everything costs money up front so here are a couple of acknowledgements. The Society of Actuaries, of course, was the one that sponsored that 401(l) program that we're doing here for continuing education. Since I'm in academia as well as in consulting, I also went to the Institute for Academic Technology, which is an IBM-sponsored institute for people in academia.

In multimedia work, you need people in different disciplines. It doesn't do just to have actuaries. It doesn't do just to have programmers. For example, Bill Gibbs has a Ph.D. in teaching computers. That's a new generation of people. When he looks at software, he is not impressed with how long it took to put it together. He's more impressed with how long it takes to use it. I think that you need people like that if you're going to be looking at multimedia work. We also run students through our systems. In our case we use students that could be in the actuarial program, but they don't have to be actuarial students, just any kid walking down the hall is fair game. "Come in and try a piece of software." If they can't use it, your software is no good. If you think it is, then let's see you market it. We are very concerned with the users.

What are we talking about? Let me give you some examples. We'll start with a basic one. The very first one that people have used, of course, is just basic concepts. I want to get an idea across. One of the things that you can do is scroll which we've all seen. I can highlight things in my picture. This is nothing more than just putting a screen on the blackboard, except that I can highlight things. That's helpful. In addition to using this in the classroom or in a teaching environment or in an instructional environment, people can use it on their own.

The next phase of this thing is our old friend the time diagram or time line. How does a time line work? Well, I have a very complicated problem. I'm investing $100 and then it's going to grow to $105. You can take a look at the equation, and if I highlight these things just as before, I can also scroll to get the same kind of thing as we did before. What makes this a little bit different is that I can do a little bit of a sensitivity study. I can do this all with a mouse and that's very helpful, because my writing is terrible. For example, if I wanted to choose, say, 1%, I change this value to 1%, run my calculation, and my equation changes. The point is, you could use this in a number of ways -- it is a very basic idea, but the difference here is the communications aspect. You don't have to be a computer person to use this thing.

The next step that we might look at is dynamic tables. Of course, everybody uses tables, but the interesting part of a table is the "what if" part.

There is the standard amortization table that all actuarial people have to know about. We can, for example, change the interest rate. Let's make it 1%, instead of the original 5%; then we can calculate. Now people can do "what if" studies. They can see what the difference is and become a little bit knowledgeable in how these things work. You can do it for whatever you're interested in looking at. You can do this kind of analysis for any table that you have.

What else can we do? How about some analysis. Suppose that you know some things but you really don't understand how they work. For example, some people are not familiar with how present values work. Take a dollar invested at time 1, a dollar invested at time 2, and a dollar invested at time 3, bring them all back, add them all up. It's the present value of the stream. You can tell anybody about this. They don't have to be an actuary or even like math.

You can have a description for them such as what this is all about. If they're non-quantitative people, or if they're people who do not like the numbers, you stop. Suppose you have more sophisticated people. You can give them a more complicated representation – a general case. You can give them the analytical form, see what it looks like. We can give them the computational form, see what that looks like. Now, it doesn't matter which one you start with. You still have the question -- what happens if you change the parameters? The interest rate goes up but what's that going to do? If a term changes, what's that going to do in this simple case? We can change the interest rate to 6% and then 15%. What have we got here? Change the term. You have 20 years, make it 30. Let's see the picture. You can do a sensitivity study on any kind of functions. And recall now, I'm just using this little 25 megahertz 386SX.

What else can you do? I'd like to talk actuarial to my people, but they don't understand it when I talk actuarial to them. I wonder if I can train them? I'm not going to sit down and train all my people. I could spend the time, and then they might move away. Maybe we can do it automatically? For example, suppose that I'm looking at annuities and now I want to talk about present value. Just a minute -- maybe you don't understand those things. Let me give you some definitions to help out. This is an annuity certain. Here's what the present value is. Let's make this payable once a year, payment at the end of the year. I'm looking at a flow chart. Here's what the "a" means. Here's what this little funny thing means, the "angle n." And here's the "n" and here's the "i." If you'd like to go down the list that you had on the other side, here's what it is. I can teach these actuarial things.

Of course, if you do it this way there's a problem. You'll never know if they learned it. Let's run the quiz. Of course, there's two things to be done here. One, I can give them the symbol and they can describe it. We all know from our own experience, that's the easy way. Then the next one is, of course, to construct the symbol given the idea. Let's see how this thing would work. That is the symbol, that's no big deal. Here's our annuity, an accumulated value, payable continuously (so it's payable more than once a year). It's not increasing; it's constant. It's not deferred. Let's check it out. There they are. How about that?

The other one though is more interesting. Ask yourself, how would you construct one that constructs the symbol? Since you are programmers, the answer is kind of obvious on reflection, or perhaps even before reflection. You need some tools. You have to imagine the numbers. We have annuity certain, payable at the beginning, more than once a year. It's a decreasing function. And it's a mature function. With any luck, I've gotten the thing. Let's give it a try. You keep track of how this worked out, and by the time kids get through this thing they should have learned the process.

You can do all kinds of things with this. If you like animation, of course, you can do some animation. How about this issue? This thing is spinning in space and the question is, how will it land? Will it land straight up? Will it land on its side? We all know that it depends on your information.

Multimedia will have as big an impact as the microcomputer had on industry, on education, on instruction. It's a fun area.

MR. ROTHMAN: There is a unifying thread; technology has to be effective. It has to be appropriate to the mission that you have, it has to be appropriate to the job you're trying to sell. You have to know what you're trying to do or you'll just use technology for technology's sake. Third, it has to be used. You can't just build a piece of hardware or software and expect that people are going to pick it up, use it and never bother you again.

MR. DONEL C. KELLEY: This question is for John. I read a couple of articles similar to your presentation with the same conclusion that productivity was not increasing with the increase in computer technology. Two things stood out. First, they held up the banking industry as an exception where productivity did increase with the increase in technology. The second thing was that they warned heavily against having so

many office experts that destroyed productivity. I wonder if you would comment on those two things.

MR. SARDELIS: As far as the banking industry goes, we also have a practice that consults with the banking and financial institutions. Frankly, that's not been our experience generally. While the banks are heavily automated, in my view, they haven't done or haven't achieved the productivity levels that you suggest. As far as the office experts go, I tend to agree with that. This gets to a question of your architecture. One of the gentlemen I work with advised me that there is a very easy way to get rich in this business -- form two companies. One is called "Centralized Consulting" and the other is called "Decentralized Consulting." His point was that you're going to be busy all the time. I think that's basically true. The real question revolves around architecture. How do you really want to create the architecture to run your business, irrespective of whether it's decentralized or centralized. The critical issue here is one of strategy. You have to keep control of the business centrally, irrespective of how you distribute your hardware. Information is too valuable. What about security? There are so many issues that come with information that require you to manage it so closely that you have to establish standards, so I agree with this idea that office experts are a problem and it's mostly the problem of decentralized organization. But what you buy there is control, so when you want to do something, you have the ability to do it quickly but the corporation at the top level loses some control and some continuity and some standards of approach.

MR. PETER L. HUTCHINGS: One of the dilemmas that you have looking at this field from a management point of view is that as a Fellow with no personal computer expertise, I have alternate classes of advice that I can get. There are what I think of as the PC zealots who think that everything can be done on a PC and they're sometimes right and sometimes wrong. Then, there are the mainframe bigots who think that everything should be on a mainframe and they're sometimes right and sometimes wrong. There aren't too many people who can cover both sides. Since it looks as though there are plenty of cases where you need the one approach and plenty of cases where you need the other, if I could give one thought to the centralized consulting and decentralized consulting, it would be that a company that would get a lot of business would be in mainframe and PC consulting as opposed to one or the other. I would be very interested in anything the panel might have to say on how to weave through these alternatives and come out with the correct answer for the particular situation at hand.

MR. SARDELIS: I can share at least some experience with you on that. There's a current situation where people have criticized the existing systems, the so-called legacy systems that exist in insurance companies. Legacy systems are those systems that have been written over the past 20-25 years that basically run the core of what we do. I had a conversation with a CIO at a large mutual and he brought up a very interesting point. We were talking about client/server architecture and whether that was something that was useful. His view was that it was useful; however, the notion that you could migrate all 25 years of code off a mainframe onto a client/server or any other platform presented me with a problem that I think is a stunner. His point was this: over those 20-25 years, so many products that we no longer sell are being administered there. Should we rewrite code that administers products that we no longer sell? I think that brings home a very important point.

The mainframe has a role. The PC has a role. The workstation has a role. How you integrate and mix and optimize that formula for your particular organization becomes your strategy. I agree with you. I think there's a lot of confusion.

Any number of consultants will suggest that the workstation or the PC can replace everything that's out there. I don't believe our industry can. We're too heavily invested. What it can do is leverage that information. I think you're going to come to an era where we're going to look to exploit what we've done on the mainframe, rather than replace what we've done on the mainframe, and use it to gain some business advantage, and use the other technologies, both the hardware and the software, in smarter ways. This idea of system integration is an important concept and I believe that the whole issue here is one of not trying to get rid of the mainframe or those COBOL applications, but finding a way to intelligently mine and exploit that information with the evolving technologies that are much more friendly and applicable in today's business world.

MR. JAMES F. HALL: I have done very well as an actuary with my computer background. I've never gotten a bad review. I've done extremely well for somebody with my exam level. I have managed what I would call IS people and I have managed actuaries, and there is just so much confusion out there and technology moves forward so rapidly that the people who generally run insurance companies really don't have a clue as to what they can and can't do. I think that's the bigger problem. I find that when they invest money in PCs, particularly, they don't want to invest enough to really train the people. As you say, it's a real problem and something that's going to take a long time to work out. I know in my own company, the systems people aren't even up to speed on what can be done and I work for a small company. Like I say, I've done well because I've taken the time and it's something that I'm interested in. Not everybody can be an expert. There is just so much out there and I don't know what the solution is but I think a lot of people are missing the boat on what they can be doing and particularly what their actuaries can be doing if they give them the opportunity and give them access to these tools. A lot of the older actuaries, who don't have the time, don't want to let their people try this stuff. In order to do it, you have to give somebody a shot.

MR. SARDELIS: I'd just like to offer one comment. I know for a long time the mainframe bigots hoarded the information; they just didn't even want to download it to a PC. An actuary could somehow do something almost perverse with it. That's changed. That mentality of protecting and job security is breaking down thankfully and we're starting to see a much richer combination of actuaries moving into information systems and information systems people starting to empathize and understand what the actuaries really need. It's a bigger problem in that information technology just takes a long time to really diffuse to the organization. There was a study that stated that, on average, an innovation takes about 19 years going from invention to production. A lot of what's out there is intermediate and many companies have been burned by getting involved too soon, so they're a little gun-shy of getting involved, but the technologies are certainly the lead technologies, but that doesn't mean there isn't a whole host of things out there that can be done. There is some question about whether management has defaulted on this issue. One of the observations done by Gary Loveman at MIT is that information technology does amplify management, so if you are a bad manager, maybe it's best to avoid it.

MR. ROTHMAN. Let me just add a comment on that. One of our roles is to manage expectations. It's very easy to become enamored with the technology. Our users often think of our equipment as toys, expensive toys, as an unnecessary expense, because we haven't explained what they are going to be used for or how they are going to help. I'll throw out four words that will probably sum up the entire problem and they are "artificial intelligence" and "expert systems." We put a lot of stock into what expert systems we're going to do for the insurance industry, for actuarial work, for everything else 10 years ago and now about the only good application of expert systems that I've seen, other than underwriting systems, is the autofocus mechanism in my camera.

MR. POLLACK: I know of one bad story. An affiliate of our company created quite a pension valuation system. Actually, they're still in the midst of creating it, based on an expert artificial intelligence concept. It has cost an incredible amount of money. They had a real tough time. I think if they ever do get it off the ground it could work, but certainly it was a leap in technology that I don't think they were prepared for. I know they haven't carried through with it yet, and I think it's just too big a leap for their applications.

MR. SARDELIS: I think expert systems 10 years ago would try to organize around a technology rather than reorganizing first and then applying a technology. It was once again heralded as the second coming. It's useful. It can have its productivity if it's properly structured inside the organization and if you organize around that particular technology and add other combinations of technology, you can do good things. But standing alone, if you're looking for the miracle or the deliverance, you're not going to find it normally with any one given technology.

MR. KERRY A. KRANTZ: Would someone like to comment on the question of Windows-NT versus OS/2 and also the potential for the Pentium computer?

MR. POLLACK: The Pentium is going to be generally stunning. I think you're going to be buying it. If you're buying 486s today, you'll be buying Pentiums a year from now. The prices will come down and I think you'll want to have one. If you really want to know Windows-NT versus OS/2 and those sorts of things, my comment would be real simple. I won't tell you my conclusion; I think there are 35,000 beta versions of Windows-NT out there and only half of those people create applications. There will probably be more applications for Windows-NT than there are users of the other systems.

MR. SARDELIS: I agree with that. Windows-NT is the de facto standard until proven otherwise. The application developers develop packages that, when available, will really create the standard, and until Microsoft does something incredibly stupid, it seems like NT certainly has the advantage.