

# THOMAS M. NIES

## ORAL HISTORY

---

COMPUTERWORLD HONORS PROGRAM  
INTERNATIONAL ARCHIVES

---

Transcript of a Video History Interview with  
Thomas M. Nies  
Founder & Chairman,  
Cincom Systems

Interviewer: David K. Allison (DKA)  
Division of Computers, Information & Society  
National Museum of American History, Smithsonian  
Institution

Date: 1995

Location: Cincom Systems Headquarters  
Cincinnati, Ohio

DKA: Let's go back and talk some about your upbringing and education. You mentioned that you were raised in Cincinnati, but tell me what it was like to grow up here, and what led you into the software industry.

TMN: Well, I was born and raised in Cincinnati. I love it; it's a wonderful place to grow up. Cincinnati is a conservative city, and I was formally educated in the conservative traditions. But the conservative traditions also realize that it's important to have good doses of the hard sciences, so in high school I was trained side by side in theology along with chemistry, physics, and six years of mathematics. I was involved in the Classics; therefore, I studied four years of Latin, a couple of years of French, ancient Greek history and things of that nature, and carried that on into college. Since I had already been formally trained in a lot of the foundational topics such as ethics and logic, in college I concentrated more on the things I thought would broaden my education in preparation for going into business. So I received an undergraduate degree in marketing and a graduate degree in finance. I felt that this would prepare me with some formal training to start in business.

But I also enjoyed all the things that any young kid did -- playing knothole baseball, football, fell in love with golf. In those days, the economy was such that you could work and put yourself through school. So I caddied and things like that to help pay for my education. And since I came from a family of six kids with a working man for a father, I did help to put myself through school, as our other brothers and sisters did. It was a really good foundation for going into the business economy later on.

DKA: Let me ask about some of the variety of your background. You mentioned that you studied a lot of things other than science and engineering. Do you find that some of those other topics have been important to you in your business career, in addition to the things that you learned in economics, science, and mathematics?

TMN: I think they are of immense importance to me, because quite frankly a business doesn't run around technology. A business runs around people and relationships, around what moves people, what teams people together, and so on. These things have very little to do with technology, geometry, physics, chemistry or anything else. They have to do with how people think, what they believe in, how they behave, how they relate to one another, what their ideals and goals are, and so on. This is really the primary emphasis of management - and not just senior management, but management of a team, management of a department, management of a group, management of a relationship of people who are serving and supporting a customer base. And so this formalized training that I had in these so-called "soft sciences" or the Classics probably has done more to prepare me for business than all of the hard sciences which I've had.

DKA: So you must be concerned that these kinds of educational values are not necessarily pushed by departments of education any more.

TMN: I think it is a great weakness in the American educational system that they have in effect downgraded the areas of philosophy, theology, logic, ethics, and especially morality. These studies help to form people who can properly lead, guide, and shape organizations. Today, education emphasizes not so much education, but training.

We train accountants, we train chemists, we train physicists, but we don't really fully and formally *educate* people for proper roles of leadership in society except in a very few elitist-type schools. These schools still see the importance of this type of education, and many people who understand it also make sure that their children attend these type of schools. I think America is suffering from this emphasis on training instead of true education, and I believe it's only a matter of time until we move back to broadening the educational experience and perspective of people -- especially the ones we want to become leaders.

DKA: Now, you did your college and graduate work also here in this area?

TMN: Yes. I have both undergraduate and graduate degrees from the University of Cincinnati.

DKA: Did you ever consider taking off at that phase of your life, or did you want to stay in this area? Was it economics that kept you here, or your desire as well?

TMN: It was purely economics that kept me here. Quite frankly, I probably would have liked to have gone to Xavier, which was a local Catholic university, because I already liked theology and philosophy, and the Jesuit university was very, very well known for that. But the tuition and other expenses were prohibitive, and the University of Cincinnati was a Cincinnati college at the time. They also had a cooperative educational program where I could go to school and study for eight weeks, and then work for eight weeks. So the University of Cincinnati was perfect for me from an economic standpoint, but I would have preferred to have focused more on the courses I fell in love with in high school.

DKA: What did you think you were going to do when you were in college? Did you have a specific goal?

TMN: I knew I wanted to go into business, and I wanted to eventually become a manager, hopefully become an executive if possible. I knew I would have a business career of some type, but in those days there was no such thing as a computer industry, and high tech was not talked about much. Insofar as the company that I wanted to work for, it was probably Procter & Gamble (P&G), because they were the biggest, most prestigious local Cincinnati firm, and I was a Cincinnati boy. It's important to remember that during the days I was in college, I had never traveled out of the city. In those times it was different. There weren't any jets, and television wasn't really coming into the scene the way it is today. Our orientation to the world at large was the movies, and basically that was it.

DKA: So as you came out of school, what happened to you? How did your career begin to unfold?

TMN: Well, there were several happy coincidences that occurred. How I wound up in the computer business was really a serendipitous event. I had always thought about going to work for Procter & Gamble, and so I had interviewed with them and was fortunate enough to receive a job offer. There were very few students in our class who were given job offers from P&G -- most were made to graduates from the eastern schools. I was so enthusiastic about my job offer that I was telling my fellow classmates about it, not realizing that some of them had not received offers, as well.

I certainly was not befriending them in any way by telling them this, and one of them finally said, "Well, why don't you see if you can get a job with IBM?" And I said, "IBM? Who's IBM?" I'd never even heard of them! Punch cards? I didn't know anything about this in 1960. In reflection, I realize that they were interested in seeing me snubbed by IBM. So I interviewed there, and as fortune would have it, they offered me a job as well. At this point, I went to visit with a couple of ladies who had helped to raise me as a young boy, and I wanted to let them know I had this great dilemma. I had this job offer from IBM who I had just heard about but now realized was a prestigious company, as well as one from Procter & Gamble, who everyone in Cincinnati knew was prestigious. I asked them what I should do. They listened carefully, and then said, "Well, Tom, Procter & Gamble is a wonderful company and I'm sure you could do well there, but computers are the future." And so following their advice, I went to work for IBM. And that's how I got into the computer industry.

DKA: Would you have been doing pretty much the same thing for either company? Were you going to be in the sales and marketing, or were they vastly different forks in terms of where they were pointing?

TMN: In both I would have been in the area of sales and marketing, but of course with Procter & Gamble I would have been selling consumer products, whereas with IBM I was selling technology. So with IBM, it was more personal selling and specific account development, whereas with Procter & Gamble it was mass merchandising and mass marketing. Interestingly enough, these two industries have now come together. The computer industry has since become a mass merchandising industry, and many of us who grew up with personal selling find it difficult to change our mindset to one necessary for mass merchandising. Microsoft, for example, recently hired Robert Herbold, a senior executive from Procter & Gamble, as their chief operating officer. Wanting to bring mass merchandising out a decade ago, Apple Computer did the same thing when they brought John Sculley in from Pepsi Cola. So it is a mass merchandising industry now, more than we realize. The worlds that were so radically different then have become very similar today.

DKA: What was it like to go to work at IBM in those times?

TMN: Oh, it was fantastic, absolutely fantastic. Some of the happiest times in my life were those first few years with IBM. It was a whole new world for me. When I was in high school, I was part of a selected group of students drawn from all the schools in the Greater Cincinnati Archdiocese and Latin School. They had what they felt were the "best and the brightest," and we were formally trained that way. I found this stimulated the best in us boys.

When I got into IBM, the selectivity of the way they recruited was such that even if you were at the top of your class in college, you might struggle just to be average in the formal training programs in many cases, because the quality of the competition and the insights and the knowledge of so many of your peers was so great that it simulated you. IBM was on a huge roll then. They were growing rapidly and they had the "pick of the litter," if you will, in college graduates. So I worked with a lot of very, very gifted, bright, intelligent, dedicated people.

And in those days, we were still oriented around the idea from Tom Watson, Sr., that the customer is king, and our job was to service that customer. We had a slogan -- IBM Means Service -- and everything was focused on this. It was a tremendous introduction. I came into IBM at the time when we still had vacuum tube computers and wired panels and plug boards, so I was sort of transcending two eras. I had one foot in the "old IBM" era - the one built under Tom Watson, Sr., who had been there as a patriarch for so many years, and one foot in the "new IBM" era under his son, Tom Watson, Jr., who was rapidly transforming the company. It was a unique opportunity, and it was a wonderful one.

DKA: What were you in particular doing? Were you here in Cincinnati selling? Did you move around the country? What was your work?

TMN: Well, IBM had a formalized training program. In those days they believed that you had to be rooted in the technology first, so for the first 18 to 24 months you were a so-called systems engineer or a technician, where you worked side-by-side with people who were permanently dedicated technicians. You did the same type work they did, and went through the same formal classes and training in order to have firm roots in the technology. Then they would move those who were going into marketing and sales into a specialized training program, and then they would put you into various industries.

After serving the first 18 to 24 months in these technical roles, I moved into the insurance industry and was formally trained in insurance at Hartford. There I was taught the end-user's perspective -- what they did, what their applications were, and how we could apply computers to that particular field. The thinking at IBM in those days was that once you learned the customer's requirements, you could use computers, which the customer knew little about, to solve those requirements. It was a solution-oriented kind of thinking using high technology. Later, it became more focused on high technology, but in those days it was really oriented around solutions. And so we did this by industry. I was in the insurance industry, but later I felt that it might be to my advantage to broaden myself and to be a better IBM person if I also could learn something about the manufacturing industries. So after serving a couple or three years in insurance, I asked to get into manufacturing. Eventually I became a large account manager for ARMCO Steel (now AK Steel), which is still one of the biggest manufacturing firms in the Cincinnati area today. That was my role and my function while I was with IBM.

DKA: Some people said IBM meant "I've been moved." Did you end up being moved around the company, or around the country?

TMN: In fact, one of the reasons I left IBM was because I didn't want to get into that particular pattern. That was IBM -- they felt it was difficult to promote people in place. In other words, you couldn't become a manager of the same people who you had formerly been working with. So they would take a fellow from Cincinnati and move him to Cleveland and promote him, and bring a guy from Cleveland into Cincinnati. They were probably correct in that, but it is difficult for young people to grow and develop that way.

IBM was in a vibrant, rapidly, dynamically growing time, so it was true that, at that time, perhaps IBM did mean "I've been moved." And if you wanted your career to move ahead fast, the idea was to put those moves as close together as you could in order to get promoted. It usually meant a geographic move as well. If you were languishing in a job for more than 24 months you were on the slow track. Everybody wanted to be promoted in 18 months. So thinking that through, I felt that if I stayed with IBM it would mean relocating not only myself but my family time after time after time. I had gone through five different grade schools when I was a boy growing up, and while in retrospect I believe that was good for me, I wanted my kids to have roots, a family, and a neighborhood. We had several children already, and our oldest was just getting ready to go into the first grade. Although I thought I had a very good career ahead of me, after talking with my wife I wondered what the moving would do to our family. My decision to leave IBM was in part because I did not want to get on the treadmill of relocating my family time after time.

DKA: Before we talk about all the factors that led to your leaving the company, tell me a little bit about the values you learned at IBM. You talked about two sets of values, the old IBM and the new IBM. As you look back, what do you see that you learned to take, and learned not to take from that experience as you went on?

TMN: Well, let me speak first about the old IBM - that's not to say that the new IBM is radically different, but just let me begin there. The old IBM was heavily oriented around the importance of the individual -- that the individual made a big difference. They wanted to stimulate individual excellence and to move the people along to be as good as they could, as individual producers, as rapidly as they could. They were also heavily oriented around the idea of developing, servicing and supporting the customer. Remember, they were still trying to gain market share in those days, and so the emphasis on selling and managing sales cycles, and so forth, was very strong, and the importance of people was promoted heavily by Tom Watson, Sr. He created a society of like-minded souls who believed along certain lines. And he believed very much in a homogenous world, if you would, and I appreciated a lot of the advantages of that. I liked it very much, since I was still a youngster, and the idea of individual excellence and individual brilliance was still strongly in my mind.

Later, as IBM moved into the world of bigger systems and bigger computers and so forth, the individual effort no longer was seen as able to satisfy all of the requirements. So we moved away from individual excellence to teamwork, team emphasis, and team effort. We'd have technicians and salesmen and managers and people in from White Plains and Poughkeepsie and Endicott, all focusing together on a larger problem. We would bring together the "team" of IBM, and you used this team capability to overwhelm the individual competitor who was still operating on a very limited resource - who had maybe only one man or two on the account. We might have eight, ten, or twelve people teamed together to satisfy those requirements. And so I saw then not only the superiority of the teamwork that was involved in competition and satisfying customer requirements, but also the need to organize your thinking more around what it is that the other person wants and needs. How do we help to achieve a team goal, where every individual also feels that he's making a major contribution and is an important part of that team? This was one of the major differences that I saw in the old and the new IBM.

But a problem I perceived with the new IBM was that, along with the growth and wealth that developed, IBM also developed into a heavily politicized environment, especially in the late sixties and early seventies. As a newer IBM person, this environment may well have been there all long, but I didn't see it in the early sixties. So it seemed to me that the growth of IBM also brought with it a political quagmire -- one which I now believe harmed IBM in later years, but even at the time I didn't care for. I wanted to do the job and to focus on the customer the way I'd been trained -- to solve problems, and not get involved in the political machinations that were, in my mind, developing at that time. So this was a part of the liability I saw that came with the growth of IBM from an internal perspective. It became not so much what you were able to do, but rather who you knew, and how your manager viewed you, and who you were connected with and so forth. And that, I felt, was a liability for IBM. Those were basically the major differences between the "old" and the "new," and I think they just came naturally with the dynamics of growth. There most likely wasn't much anyone could do about it. It was probably managed as well as it could have been within IBM.

DKA: So as you looked around in this late sixties period, there were a number of factors that began to make you think that maybe you would consider something else. You mentioned the moving aspect. You talked about some of the ways the company was developing. Were there other things that began to impinge on your thoughts about what your future was going to be?

TMN: Oh, quite dramatically so. The moving actually was an ancillary issue -- the big factor was that I, like all other IBM people of my year, was in love with the hardware. We could quote the fees and the speeds and the prices and everything else of this equipment. It was the world of big iron. That's what we knew. That's what we loved. We were devoted to it, passionately so. It was a world of fabulous lights on the processing consoles -- the kind of things we used to see in the old movies -- reels of tape going and big rooms filled with computers and air conditioning units and raised floors. It was quite a splendid and very impressive environment that we all loved.

While I was account manager for ARMCO Steel, we installed one of the very first online system IBM 360's ever. And in those days there was no software of any type for this environment, so it fell to us to provide the software for it. I became the project manager for a set of software development efforts, which included people from New York and other places, to implement the software that was necessary for this computer to do its work. Now this entire time, the computer was sitting there doing nothing until we finally got that software system working and operational -- at that point in time we gave life to the whole thing. And I saw from working with it intensively, month after month after month, sleeping in the data centers and so forth around the clock, 24-hours-a-day, that it was the *software* that actually made all the difference.

So it was a tremendous conversion for me, from being an advocate of hardware, and a great believer in hardware, to seeing that without the software, the computer couldn't even generate heat. I saw that the software was everything. At that time, I began to promote to IBM, from my vantage point, that software was the future. Not unlike those ladies who told me some years earlier that computers were the future, I was telling IBM that software is where it's at. It's the software that gives life to the computer.

Well, IBM's point of view in those days was that the problems were in the software; the money was in the hardware. We wanted to push as much of the problem of building, developing, and maintaining software off onto the user as we could. We promoted to them that they should develop self-sufficiency, develop their skills, and learn more about software. In part they wanted to help themselves, but it was also true in part that IBM didn't want to have to do all that work for them.

Remember, in the old era, you paid for the hardware, and the personal service was bundled, or included in that price. As we moved from the old days of punch card accounting systems and wired panels to computers and stored programs, software rapidly became a very big requirement to get those computers to work. IBM in effect was throwing in all that service, and so there were lots of problems for IBM. It was natural that they saw a huge drain of resources by software away from the hardware revenues and profitability. And so their mindset in those days was, the money is in the iron, the problem's in the software. And the more I talked about software's opportunity areas, the more they said this was completely wrong -- software is a big problem.

But I had been trained by IBM to see that the opportunity is in solving big problems. And moreover I had now fallen out of love with hardware and in love with software. But the more I talked with IBM, the more they seemed to be absolutely adamant against it. Therefore, I felt that, even though there was no software industry at the time, I would probably have to leave IBM and attempt to find a software company to work for. Eventually, I grudgingly and sadly came to that final conclusion. I left what might have been a very good and wonderful career for me, had I not possessed that chance event of being in charge of that software development effort. The primary reason I left IBM was because I had seen the potential of software, and had fallen in love with it.

DKA: How did this look from a business perspective to IBM?

TMN: Well, of course, I wasn't privy to the machinations of the upper reaches of IBM thinking, but in retrospect I can imagine what they must have thought. Here I am, talking about the importance of software and the great possibilities it had, when concurrently, we had the IBM operating system and the tremendous cost overruns of that project. The tremendous difficulties of getting it completed prevented IBM from delivering, installing and picking up rental revenue on tens of millions of dollars worth of equipment. So I can see why they would see software as a huge liability, and a terrible problem.

In retrospect also, or even somewhat at the time, it began to be seen that the software drains were actually depleting the huge economic resources of IBM and were possibly bringing them to the point of illiquidity. Certainly in the field they were draining technical resources of every type imaginable, sending them off to Poughkeepsie and Endicott and other places like that to help in the software development efforts, which were floundering.

And so it was easy to see that IBM would have a very jaundiced view of the potential of software -- especially when their operating system was delivered as a bundled part of the equipment, and there was no revenue from it -- yet there was a huge amount of cost. And that operating system, as important as it was to keep many jobs running concurrently in the computer, and managing all those powerful computer resources, was just a small part of the overall software requirement that had to eventually be developed and delivered. So I could see where my views would not be too popular within IBM at the time.

DKA: And you mentioned you were thinking about maybe going to work for a software firm, but it's not like there was a software industry out there to work for, or a lot of choices. What in particular did you really think you were going to do as you thought about going some place else?

TMN: Well, I use the term "software firm" today, but in those days it would really have been a professional services bureau or something like that, because packaged software certainly was not an industry. There might have been a couple of firms around at that time, but there weren't software firms in every city. In fact, there were so few of them that as I began looking around, I found there were none in Cincinnati. So I wound up actually starting a software firm myself.

My objective was never to be a so-called entrepreneur, or founder of a company. I wound up doing so almost by default, as the only way to have a job in a field that I wanted to work in. When we started our firm we had no software products, we had no customers, we had no real idea of packaged products as an industry possibility because there was no historical precedent for this. So we began by offering a personal service - "body shop" programming -- whatever we could do to help our clients implement the computers they were installing.

DKA: Well, I want to stop you there because the way you've described your background and career, you don't sound like the kind of person who says, "Let's go form a company." Describe the environment when you left this major, stable company to go out on your own with something that wasn't being done elsewhere. That's a big decision for a man with children and a stable background.

TMN: Yes. Well, I tell you, it was an interesting situation. When I joined IBM, we didn't have two nickels to rub together. Remember, I'd gone through many years of college, married, and we had our first child by that time. We were absolutely broke. I like to say broke is a temporary situation, whereas poor is a permanent state of mind. But after a number of years with IBM, we had now earned enough money to buy a house, to have a nice car, to have money in the bank, and to be able to take a little risk. I told my wife, "Look, just a few years ago we were absolutely dead broke. Now we've got a home and everything else, yet we could lose it all and be broke again. But that's temporary. Now that I have this background, this knowledge, this education, I could go to work for another computer company, or perhaps even IBM would hire me back." So I could always start again if it didn't work out the way we hoped it might.

I had only one career, and I had to do something with that career that I loved doing. If I was doing something I no longer believed in, I didn't think I could ever do a good job at it. And I believed, truly believed, in software. The risk and the unfortunate circumstance of leaving all my friends, my colleagues, the social environment, everything that was IBM was a very difficult choice for me. Very difficult. Starting up literally all by myself was not something I looked forward to, but I really did believe that software would be a major industry in the future. So we began.

DKA: When you say we, who are you speaking of? You didn't go off totally on your own, did you? What was the early formation of the company?

TMN: Well, I did literally go off on my own, but I told a few friends that I was leaving IBM. And when I told them, they said, "Oh, we'd like to join you." So a couple of other guys also left IBM. We started the company and decided we had to have officers to identify who was going to do what job. They said, "Well, you're the president, and we're the technical people." And that was basically how we got started. I had the title of president, but I was primarily the salesman. We sold technical support to customers. At first, there was just myself and two other guys, and within a few weeks time I had sold 100 percent of our inventory. So they were on jobs and on contracts, and all of a sudden I had nothing to do. So I thought, "Well, this is not a good business model." The solution was to hire more technicians so I'd have more resources to sell. So I was then recruiting technicians and selling them into the various accounts to help implement all these new computers that were deficient in software.

As it turned out, our clients were primarily having problems with data management, and we found we were solving the same problems over and over and over again. We became very good at data management. We understood what the problems were. So I called together our people one day and said, "You know, instead of solving these problems over and over again for different customers in different ways at a very high cost per customer, is it possible that we could somehow develop some programs that I could sell? They thought about it for a while and said, "Well, it's not what we do, but maybe we can."

So we all wound up working days for the clients we had on contract, which paid the bills, and at night we had a research and development effort going on. We were developing computer programs till 2:00 or 3:00 in the morning, and we'd be back on the job by 7:30 or 8:00 working with our clients. Basically that's what we did. Our nightly R&D team was the same team of people who were working with clients during the day. We self-funded our R&D in that way. We didn't have an "angel" working for us who funded the development of the data management projects. We built the systems working in the evening until we had some rudimentary products that we began introducing into the marketplace in 1969.

DKA: Let me see if I understand correctly. Initially, you started off almost as a services company, solving computer problems but in a service type of mode, and then you began slowly to develop your own commercial software products. That's a choice of sorts, because it's quite easy to stay in the services market without developing your own product.

TMN: It was a choice we made primarily because I was looking to make more of a contribution. Hiring people and putting them on contract was almost a part-time job for me, and I wanted to be more meaningfully and gainfully contributing to the firm. I thought if I had a package, a product of some type, I could continue to sell that product even though I didn't have technical resources -- when all of our technical inventory was consumed working on projects for clients, I could be selling product. So it wasn't a choice of either/or, it was a decision to do both. Our technicians were going to continue to do implementations for customers, but I would sell products to supplement our revenue sources and also to deliver a lower cost, more efficient packaged solution to the customer.

We did make a choice not so long thereafter, though. What happened was that since the package sold for one-tenth the cost of most of the service contracts, it was readily chosen by the customer time after time. So we were able to sell a lot more packages than implementation contracts. Once that began to happen, we said, look, let's get out of the technical implementation business and put those resources more and more into enhancing, expanding, and helping sell the products we build. So we very rapidly moved away from being a services company into being a products company through a process of discovery.

We discovered an industry. There were no other companies doing this at that time. We literally helped to create the entire database industry through this process of seeing what the customer problems were and responding to how we could help solve them. You have to remember, this was in the late sixties. IBM wasn't selling software. There were no other companies. It was an interesting situation, but that's how we wound up in the software products business.

DKA: Can you be a little bit more specific about who your early customers were?

TMN: Well, at first our customers were within thirty or forty miles of Cincinnati and were the smaller or more moderate sized companies. However, one of our earliest clients, Hillenbrand Industries (near Indianapolis), is a very large company now. Then in the late '60's we branched out from Cincinnati. We went to Indianapolis, where we sold Eli Lilly, which is a pretty big company. Three or four of us were competing against a team of about 40 people from IBM for that - in Detroit, we sold Chevrolet, which was a gigantic company there. We sold Dupont, and we began to get some of the more prestigious companies based on the success of the smaller to moderate-sized companies we had. Today, we have hundreds of clients who have been with us for well over 20 years.

At that time, there was a growing awareness among people that data management was the foundation they had to have, and it was important to build upon. Soon database and database management was an area that large and small companies alike had a requirement for. We continued to expand our products, working with leading companies, leading edge thinkers, adding new features and new functions that were important for them to have, and in the process solving their problem at the same time we enhanced our offering. In this way, we broadened our base.

Eventually, we decided that the companies to sell were the Fortune 1000 companies, because they had operations all over the world. If we could sell them and do well at a specific location, we could therefore sell many copies of that product around the world. In this way, we felt we would be able to expand into a global company on the backs of satisfying the requirements of these large-scale firms - and we did want to become an international company. So we continued to focus toward the large international firms, and moved on to places like Minneapolis, where we sold 3M. 3M decided to disseminate our products throughout all of Europe and gave us the capability of setting up operations in Europe, and so we had a goodly list. I kept a book of Fortune 500 companies and underlined in red the ones that were ours. At one time we had probably 50 or 60 percent of the Fortune 500 companies underlined, and we still weren't more than three or four years old. So that's basically how we developed the business model.

DKA: Tell me a little bit more about just the nature of your product. I think this is the TOTAL product. What did you call it?

TMN: We called it TOTAL because we felt that we wanted to do the total job for the customer. Our commitment was to do the total job, and we would continue to enhance, expand, and add feature and functionality so that no matter what the customer's data management problems were, TOTAL would solve it. So this was really our first system and it became, during the seventies, the best-selling database management system in the marketplace -- widely used -- a very, very successfully implemented product. We're still known today as the providers of TOTAL.

However, we next began to see that database management was only one of many problems that needed to be solved. It was like the engine of an automobile. It is only one of the requirements, and we had to have a multiplicity of technologies to really be able to solve the total problem for the customer. So later we branched out into a lot of other technology areas as well.

DKA: Let's finish this story before we talk about some of those as well. So it was basically doing business management, billing that kind of thing. What was the early functionality, and how did you choose to go out?

TMN: Basically, database management was a concept that we were the primary champions of in the early days, although there were others. For example, CODASYL had a study team with major users and a couple of hardware vendors and others working on what they called the *CODASYL Database Task Group Report*, which was defining what the requirements were for database management. General Electric had brought into the market a product called IDS. IBM was working on a system they called IMS in those days.

But essentially the great transformation was this: historically, people had built applications one application at a time. They built accounts payable, purchasing, inventory control, payroll, billing, and accounts receivable as independent applications. Then when you built that application, you defined as part of your requirements a set of data files, and in those data files would be certain information: the customer name, the customer address, customer number, other things about that, which that particular application maintained. And as you got more and more applications going, each application built its own set of data files, and so we saw developing what we call the "my file" syndrome. Each set of applications not only did application logic but maintained data files, and so we had data redundancy, where application A had one way to describe the customer, application B had a different way; application C had one set of inventory balances, application D had a different set.

This proliferation of data into every independent application being managed in its own right was built on a file-by-file basis, with applications being done pragmatically for that particular application, designed to run most efficiently for that application, and to satisfy only that application's requirements. But the proliferation of this data redundancy brought chaos into the companies. Data no longer meant anything because people would get five different reports and the inventory balances would say five different things. What were our sales during April? Well, you'd get five different numbers, depending on how you total things up. All of them would argue that they were correct, but the data itself was all wrong.

Today, quality improvement experts everywhere say that if you have good data, it speaks for itself - and conversely, the data the customer had at that time was bad, very bad. So we went to them and said, look, you must separate data management out of any application. In that way, the data management can be performed as a separate activity which we call "database management." So we changed from data files to databases, and the idea was to have a common database, whereas the inventory record would appear once, and every application would use that inventory record. Therefore, if application A would update the balances, that balance would be current for application B when it wanted to draw on inventory. This really was a quantum leap forward in the thinking process. It was a far superior way to manage and organize and control data than the old application-by-application management of data. And so database management became a tremendous area of interest as more and more people began to see what problems we were solving.

Now, in addition to that, the most complicated and difficult types of programming that people were doing in those days were not the application procedural logical types -- updating your customer record, or comparing this to that -- but actually managing and maintaining with integrity the data for the application. Every application had anywhere from 40 to perhaps 80 percent of its programming logic built around data management. Once we segregated out the programming logic from the application of the database, and we provided a program that did that, we were eliminating all of that programming logic for each and every application. So in one stroke the customer could install a database management system from Cincom which would do the work of many, many programmers in a far superior way. It was like jet engines versus propellers. They were a different type of power, and in this way, we were able to do for customers things they could never do themselves. Database management therefore became one of the foundation principles -- in order to bring integrity into the information in the company you had to go with the common database concept.

So those were the ideas we were selling over 25 years ago. Interestingly now, David, we're moving more and more to client server and workstations, where people are proliferating database management systems all over the place. As people buy these database management systems and implement on them on various diverse workstations and Unix boxes, we are regressing back to what we in effect saw as the "my file" syndrome in 1970 where the database records on my workstation may be different from the records on another workstation. They're not relating one to the other, and we are in fact moving to massive data contamination and destruction of the integrity of data all over again.

What's happening is that many of the people doing these distributed client/server implementations are making the very same mistakes as the users did in the 1960's -- the exact set of mistakes. Not everywhere, but in many places we're seeing a proliferation of very bad data implementation processes and practices which will come back to us. The only way to solve that data integrity problem is to move to distributed data, where you might have multiple recordings of data on these different boxes but with a centralized "traffic cop," whereas any time data is updated in one place it's updated everywhere. In this way we retain consistency of data. Otherwise we're regressing back to bad data and bad information. It's an interesting problem. It's going to be a great opportunity area to correct a lot of the problems that are being made in client/server and distributed database implementation systems today. But in the '60s and early '70s that was the primary architectural problem that we were attempting to solve - to bring integrity to the data. At the same time, we radically reduced the cost of implementation of applications.

DKA: The point you've just been making leads into the question of what's technically possible, not only what's architecturally sound. As I heard you talking about your goals for the 1960's, it came to my mind that many of the things that you were discussing were not easy to accomplish technically, because of the batch process oriented mode, rather than the capability for transaction mode, the capability of the stations logging in, how much you could distribute, the use of the information. So how difficult was it? What were the challenges from the technology that was available to implement the kind of architectural ideas that you were promoting?

TMN: Well, it is an interesting problem that you bring up there, because the knowledge of what can be done and what the problems are, many times is developed around the technology that's available -- you develop your thinking processes around what's technically possible. However, the technology changes so rapidly that during the implementation process whole new orders of thinking could be possible. But of course, you can't continuously throw out mid-development cycle and start over again. So even as we're implementing today, we're implementing along lines which we can show are no longer technologically appropriate.

But what's happened is that there has been a development of the mindset that "this is what's correct." This mindset is a function of a laborious process of learning through experience and training and understandings and so on. And one develops a comfort zone and a power base around an ideology or an approach, which once it's fully learned, is already obsolete and is really no longer the best thing to be done. Then what sets in is resistance to change because so much is already invested.

People have strengths in certain areas, the company has money invested, and the companies who are actually promoting those technologies are making huge amounts of money -- they certainly don't want to say, well, wait a minute, what we're promoting today is really not the best. If that's what the customers seem to want to buy, or can be easily induced to want to buy, so be it.

The issue of acceleration of the hardware and software technology to what's possible versus what customers know and understand, is such that the great bulk of customers quite often are implementing along the wrong lines. This is a serious problem, because the biggest voices in the marketplace are not from the leading-edge companies with new ideas, but from the big entrenched vendors who are promulgating ideas of the past. And this goes on year after year, decade after decade, and it just transfers to a different set of vendors who are promoting the old ideas. It's a difficult problem.

Cincom certainly has the some of the same issues. We have people who have spent many years developing skills and expertise in certain areas. It takes a continuous process of internal education and training by those who are the leading lights, the leading luminaries within our company, to continually keep us apprised of where the future is and help us all get there. And yet it's a double bind, because often to customers, the future is yet *in* the future, so there are only a few today willing to buy what will be the future. It's a complex matrix that one has to continuously sort through.

DKA: We're going to talk some about the hardware side of your business, the hardware that you implemented on in the early years of the seventies. Tell me how that worked for you as you began to develop your product, TOTAL. How did that marry with the hardware that was available?

TMN: Well, first of all the software business is a function of making very, very large commitments to building a software product and then charging each individual customer a very, very tiny share of that cost. So you have to find ways to spread that development cost across the large universal users. If you can do that, then you can make money in the software business. If you can't find very many users, you're going to lose on that particular effort. So since most of the users in those days were IBM, especially the larger scale users, we built for IBM. That was what we knew. That was our background. That's where the money was. And so we built, developed, and supported IBM in their various operating system environments.

Now, within IBM, there was a radical difference between their MVS or large scale operating system, and their DOS or VSE small scale operating system. So much so that there was no compatibility between the two, or at the very best, there was limited compatibility between the two. It was a major conversion to grow from one to the other. Since we had customers who were using both the VSE or the DOS system, and the MVS system, we wanted to minimize the conversion effort. We implemented the database approach, which as I said, implemented 70 to 80 percent of the application programming logic in such a way that it insulated the user from VSE or MVS. In this way, they could migrate much more easily from a smaller scale environment to a large scale, or they could decentralize work from the larger to the smaller scale with minimal amount of conversion.

Now, having perfected and developed our architecture around those lines, one of our customers came to us and said, look, if you can help us insulate ourselves from VSE and MVS, why can't you do the same thing for our hardware vendor? And we said, what do you mean? We couldn't believe people wanted to buy anything but IBM. I mean, we'd grown up there, we'd been trained in IBM, we thought anything but IBM was second rate material. Why would anyone want to do this? But the customer said, look, the economics is this: for us to buy Honeywell or Univac, they have to price their equipment at a much lower price, and we can save 20 to 30 percent on hardware processing costs. Hardware processing costs in those days were still very expensive. So the customer asked us if we would work with Honeywell so that both Honeywell and Cincom could jointly commit to delivering the total database system which they were using in their environment quite happily on IBM. Could we also build that for Honeywell?

And so because the customer demanded this of Honeywell as a requirement for them to buy Honeywell and throw out IBM -- Honeywell did work closely with us. Otherwise, in those days, I have no doubt that Honeywell would have turned up its nose at working so closely with a little software company -- because again, software was not what it is today. So we then began a development effort with Honeywell to build and develop and market TOTAL on Honeywell. Now we could make applications almost as compatible across Honeywell VSE and MVS as they were among IBM. The customer was able to move his application portfolio with the only concern being the differences in the operating system and a few other things, versus the tremendous differences in data management.

So by separating the data management programming out of the application, we now saw that we gave the customer not only much better integrity of data and much lower cost of implementation, but also much greater flexibility in how he could deploy these applications across a broader universe of computers. Once we were working with Honeywell, we realized that whereas in the IBM community where we had to sell *against* IBM -- our systems versus what they wanted the customer to do; with Honeywell, we could work *with* them because they would promote our software into their Honeywell community. In this way, our selling costs were greatly reduced. And if they were willing to promote us, we would happily build this technology for them, because we couldn't afford to do it for one customer or two or five. We had to get a pretty good size customer base to justify the whole logic. Honeywell, who did not have a database technology that was really widely accepted in the marketplace at the time, saw this as in its interest as well, because they couldn't go forward if they didn't have this capability. After all, who would build database systems for Honeywell, which only had maybe two or three percent of the market?

DKA: Can you give me a time frame that this happened?

TMN: This happened in 1971, '72, and so 25 years ago or so, we began to promulgate the idea of a transparency across processing platforms. And we really made that very, very, very transparent. Our technologies had almost the same feature and functionality across these diverse processing platforms. It was only a simple logical step from Honeywell then to go up the road to Dayton and talk to NCR about how they could expand their potential if they had better database technology, and to Univac, and Varian, and Digital, and many others who we began working with through the seventies to promulgate our technology.

We had probably 20 different processing platforms that we were supporting with almost identically the same database technology. TOTAL ran in all those environments for 10 or 12 years before companies like Oracle started to promote the idea of transparency across processing platforms.

DKA: You talk about this as if the strategy was clear and then it followed, but this must have been an enormously difficult thing to actually accomplish. What were some of the issues that you faced as you began to move from an IBM framework into a multi-hardware platform?

TMN: Well, it was an enormously difficult problem from an implementation standpoint, because they all had different assembly languages and different operating systems we had to interface with, and so forth. But the complexity of the problem was ours. That problem was all focused on Cincom, and so whilst we had to spend a goodly deal of time, money, and effort in building this, we actually were only transferring away, we weren't creating a new system. We were in effect using a technology that was already perfected and existed, and now we only had to cope with the differences in the programming languages and the processing platforms, and that reduced the difficulty by probably 70, 80 percent. So it was only about a 20 percent difficulty to implement a transition of technology from one environment to another, versus completely creating it.

Secondly, because the application interface to the database approach was identical from a customer's standpoint, there was virtually no cost associated with it, so the cost and the difficulty was localized to us. And it was difficult -- there's no question about it -- but the hardware vendors themselves saw it was to their advantage to work with us, so they sent technicians to work with us to help us. Together we jointly shared the cost, the difficulty, the effort, and the complexity of building to satisfy the user requirement.

DKA: Was this something that was difficult to sell internally, since it meant new work, new systems, new developments, for your team, or did your people share your vision?

TMN: It was difficult to sell internally, because again, the people believed in the IBM environment. That was their background. That was their experience. That was what we had installed, the IBM computers and so on. Many times we didn't even have an NCR or Honeywell box, and we needed that for testing and support, so it was difficult to sell. The way we were able to implement that was again a strategic, tactical approach inside the company. We developed a group, which we called our Ventures Group, which was designed around new ventures in environments other than IBM. So these people were focused around the non-IBM world, and that became every bit as important to them as the IBM world was to the rest of the company. In this process they began to see in those non-IBM computers many features, functions, and capabilities, which were better than the IBM offerings.

It was a real eye-opener for us to see other vendors who we had thought were second class offerings, in effect being real first class providers of lots of technology -- and that was appreciated by these people who were dedicated to it. That was their world, that's where they lived, and that's one of the ways we got the zeal and the ardor for it internally.

We needed that zeal in order to deliver a first-rate product, as well as the technical training, knowledge and know-how of those environments. And we got it by dedicating resources to it, concentrating them and focusing them in these non-IBM environments.

So there were a lot of pluses to it. There was also a huge minus, though, which became a very, very big problem for us. This was the fact that since it now seemed we weren't just offering an alternative to IBM, IBM began to think that Cincom was somehow or other trying to convince customers to convert or migrate away from IBM. This is something we were never trying to do, but it now set IBM up against Cincom as an enemy somehow. We were actually trying to help their customer, but IBM saw perhaps a threat to account control, some intruders into their space of account affection, and so on. It tremendously increased the competition between Cincom and IBM when we were still a very tiny company and they were a mega-giant. So that negative perception may well have hurt us more than the non-IBM affiliations helped us.

DKA: How did you make a decision as to what other vendor you would develop for? You mentioned Honeywell, of course. How did you decide? Was that your customer's decision? What was the rationale?

TMN: Well, I wish we could say there was a scientific rationale that we went through and it was a studied management decision, but basically it revolved around three or four factors. One of them was, is the market space there adequate? Is there three, four, or five percent of the market? Is there enough out there that we can sell into? Secondly, do the customers have application requirements for this? Thirdly, will that hardware vendor get behind us and promote us and work with us, or are they going to be antagonistic to us? And then fourthly, what was the magnitude of the conversion effort, and could it be readily supported over time, and could we deliver a quality value solution or not? These factors pretty much determined which ones we would support.

DKA: Now, you were managing several kinds of growth in this period. You were managing not only growth to other systems, but you were managing growth overseas, you were managing a transition of technology -- of your software, I should say, as the hardware technology began to change and develop. You had existing products, and you also mentioned that you were, in addition to database management, beginning to go to other sectors. As you look at these variety of options from your perspective, what were the highest priorities, and why did you prioritize things the way you did?

TMN: Well, you're right, David, we were a little bit like those entertainers who used to be on the Ed Sullivan Show where they would have all the plates spinning at one time and they run back and forth trying to keep them all going. We were trying to cover, and we were active on a lot of fronts that were conflicting for us and that were draining our resources. One of the highest priorities for us was building and becoming an international, worldwide marketing distribution service -- a support organization. We had to be more than a localized American provider, so that was high on our list of priorities. Secondly, we wanted to make sure we continued to dance with the one we came in with, so database and database management was very important to us, and we had to do everything we could to further that.

But it was also important that the customer requirements for additional technology areas that worked on an integrated basis with database management also were satisfied, because database alone, in time, became only part of the solution.

So it was a very active forum of continuously trying to balance all of these varying priorities, and the best way, quite frankly, that we could do that was to get all of those priorities in the mind of one man and for him to continuously think through all of these conflicting things, and in those days we ran really a company that revolved heavily around myself. I was active in the international affairs, I was active with our database group, I was active with our sales group, our marketing group, our R&D group, and so on, not unlike maybe Microsoft revolved around Bill Gates very heavily in those early days, and all of those thinking processes were constantly being fed to me and ingested by me, and I was trying to continuously prioritize and allocate resources.

And all of this with a eye to the bottom line, because we didn't have outside sources of capital willing to plunge huge amounts of money into our coffers for us to develop new opportunities. We had to in effect bootstrap our way forward, making profits and reinvesting those profits all along. Even today, banks are reluctant to loan to software companies, but in those days we were pariahs. Nobody would.

And so it was a complex problem of priority setting and balancing that probably could only have been done through a non-delegation, running around an entrepreneurial style that focused heavily around the key man, who was the president of the organization. It was good for me, though, because it required me to be intimately involved in knowing the technology. So I became pretty much of a technologist. It was very important for us to learn about marketing, sales, distribution, and services support, because that was fundamental to our business. It was important to know about international growth development marketing opportunities, and I was also managing the bottom line and becoming pretty proficient in finance and financial affairs.

So it was an education and training process that was a tremendously stimulating one for me. I benefited quite a lot from it, and I believe Cincom also benefited from it, because we were able to grow 60, 70, 80 percent per year, compounded year after year after year, all through the seventies, with no cash to work with. We introduced three or four major technologies, and that at that time we were the only ones in the entire industry promoting those specific technologies, and even certain ideas. The industry itself later followed suit four, five or six years later, with exactly the same things that we had introduced. For instance, we promulgated and promoted the idea of international expansion four, five, six, or even seven years before software companies ever thought about it.

I think all of this was possible because the president, myself, was so intimately involved in all of these affairs concurrently and saw the importance of all of them and how they interrelated. That's not to say that as we grew in size to today's company that we could still operate that way. It would be constrictive for us and difficult. But in those days, with the size, the number of things that we were trying to do and the resource we had to work with, I think that was about the only way we could have operated. Most of the other companies that were growing rapidly at the time operated in exactly the same way.

DKA: Let me change the subject here a little bit and ask about Cincinnati as the hub of your business. You said earlier why it got started here. As you began to grow and develop, did you find that it was a good location? Did it have limitations? What were the strengths and weaknesses of being in this location rather than on the West Coast or something like that?

TMN: Well, let me focus on a lot of the strengths, because there are so many advantages. First of all, even in those days, with air travel being what it was before Cincinnati became a Delta hub, which greatly improved travel out of Cincinnati, we were still only one hour by air from Detroit, one hour by air to St. Louis, one hour by air to Pittsburgh, one hour by air to Cleveland, one and one-half hours to Atlanta. We could drive to Lexington. We could drive to Louisville. We could drive to Indianapolis. And really, within this so-called "rust belt" was a heavy concentration of clients. We were no more than an hour or an hour-and-a-half away from maybe 70 or 80 percent of the manufacturing and all the support resources needed to support American manufacturing.

In retrospect, if we looked the whole map over we couldn't have picked a better city to be in than Cincinnati from a logistics and a marketing opportunity standpoint. Pittsburgh has large numbers of Fortune's top 100 companies. So does Cleveland. So does Chicago, and so on. Cincinnati has a number of them, as well. We were in the right markets, and we could fly to New York when we wanted to. We could fly to Boston when we wanted to. But we concentrated here and developed in the Midwest. That was our bread basket. That's where we grew.

Also there weren't many other alternate employers. Cincinnati was not known then, nor today, as being a software center, and so people who wanted to come into the software industry -- the best and the brightest -- flocked to Cincom. This is their home, this is where they dedicated themselves and committed themselves, so we didn't have the itinerant worker passing through. We had dedicated Cincomers, as they call themselves, who were committed to our mission. We were able to get not only the best and the brightest, but we could afford to invest heavily in them because they were with us for the long haul. That was another major, major advantage. We were also able to recruit the best and the brightest from the many nearby college campuses. They didn't have to relocate. They were here in their native environment and near their home. So the ability to secure, retain, develop and train a workforce was strong here in Cincinnati. The market opportunity was great. The cost of living was tremendously favorable compared with a place like New York or Los Angeles.

But the disadvantage we had, and it was a tremendous disadvantage, is that early on the marketing and the promotion of the industry began to centralize around Boston, with *ComputerWorld* and some of the other consulting groups and so forth up there. I'm not sure they even knew Cincinnati existed, let alone that there was a software company here, so they promoted heavily the local firms who they knew, or went to lunch with, or who were around their areas. And so from a marketing and merchandising, and a public relations standpoint, we suffered. Later, when there were a lot of software companies springing up in the Silicon Valley area in California, the same sort of a thing happened with the media, the consultancy groups, and the focus of attention there, and so they benefited heavily by marketing and merchandising.

Marketing and merchandising, public relations, and advertising and promotion, which are absolutely essential, were fields that Cincom was never much involved with or exposed to, and therefore never benefited from. That has been and continues to this day to be a liability for us, so from that standpoint we'd probably be much better situated out in the midst of the other software companies in California, or up in Boston. But on almost every other scale, Cincinnati is a better location than those. In my opinion, and I think in the opinion of many who work for Cincom, Cincinnati is the right city to be in.

DKA: You talked some about your workforce -- the dedication, the interest, the kind of people who came into the company. Talk a little bit more about the people who helped you in the early days to develop the company, take it in different directions. Who were they? What kind of people?

TMN: Well, we had a composite. Since we worked heavily in the IBM environment, it shouldn't surprise you to know that many of the early people who joined us were people who also had worked with IBM, saw what we were doing, believed in what we were doing, and also wanted to join us. They did so directly from IBM, and so probably at one time, perhaps 50 percent of our workforce in the early years was directly from IBM. Later, some of the people from our customer base who saw what we were doing with our technology, also wanted to join us. But I felt that we couldn't draw exclusively on IBM people or our customers as the single means to build and develop our company. I felt that we had to follow the same kind of concept that IBM followed, and that was heavily recruiting the best and the brightest from college campuses. So by 1972 or 1973, we were already going directly onto the college campuses and recruiting the best we could find and bringing them into Cincom.

Now, one of the problems was in those days was that there wasn't a computer sciences curriculum, or an emphasis in computer sciences like there is today. So when we would recruit, we might get a mathematics major or an economics major or a music major, and we would then bring them into training programs all summer long, intensively teaching them computers and programming and so on. We would really inculcate in them not only the ideas of computing and so forth, but the ideals that Cincom stood for, with the idea that we would be doing for them what IBM did for me -- which is make a large scale investment in them, teach them, train them, develop them in the computer field. We hoped that many of those people would believe in our company over a long period of time so that as they grew out of their twenties we would have an active, energetic, well-trained group of people who had eight, ten, or twelve years' experience with Cincom while still in their early thirties. This worked out very, very well for us. So we rapidly went to an environment from hiring experienced, trained, seasoned vets to investing in good people for the future. We had a formal education training program that was really terrific, probably was the best training program developed ever in the software industry, and today may still be the best.

We also went back into the professional services business, partly because those young people were not yet skillful enough with what we had trained them to be either developers of software or sellers or supporters of software products, so we had place them on contract with some seasoned, experienced people. In this way, they would fortify and, through work experience, learn and understand now truly what we had taught them in the classrooms.

So we went back into the professional services business not as a money-making opportunity - in fact, we were losing money on it -- but as part of the education and training program to get these people into Cincom and into the computer software field.

Now, that's basically how we evolved throughout the seventies in America. In our international business, we did not build on exactly the same model. Internationally, we didn't have the concentration of resources in any one country as we did in the United States. We had French, German, English, and so forth, so we couldn't afford to have these major training programs. We continued to hire the best and the brightest from the professional fields, and even today our international group is based more on seasoned veterans -- trained, highly skilled people who have already established themselves. We are continuing to grow our organization using experienced people.

For us, retention of staff is a fundamental idea. We want to believe not only that we can attract good people, but also train and retain them. Therefore, these three ideas all stay together -- attracting good people, training them intensively, and then having a work environment and a satisfaction that is so stimulating that this is the only place they want to work. It's a major part, even today, of our corporate strategy -- making sure we can keep top-notch people.

DKA: We talked about the establishment of Cincom in the world of large mainframe computers, the growth in that world as it began to evolve, the other computer manufacturers you developed software for. But as you moved into the seventies the computer industry was going through a fundamental change, moving from a world of large mainframes, to even the families of systems that IBM have, to a world in which the whole idea of minicomputers and distributive systems came up. How did this change of environment look from your perspective as a software company?

TMN: This was an interesting area because these were major shifts. I mean, they were quantum shifts not only in the processing costs but also in the kind of software, the approach, the nature of the buyers and everything else. I mentioned earlier that one of the things that we had done was to begin working with a variety of different non-IBM vendors, and this helped us to stay very close to what it was that these people were providing and what the economics were and so on. Remember that we had this special group of people who lived around these computers and became very closely involved with them. As they began to promote to us the advances and the price performance and opportunities in computers like the PEP, for example, we began to see that, hey, this computer can do everything that the mainframe can do. Or almost everything, but at a significant lower cost, and is more deployable throughout the organization, with a lower cost of support and everything else. We wanted to do what we could to begin getting in this, but it was a matter of economics.

Eventually, the dominance of the IBM influence in the account would have to also give way to economic considerations, and politics would follow economics in the user community just like it does in elections. We were, quite frankly, mainframe players. Just like Cullinet, Software AG, MSA, ADR, and Pansophic -- the companies that built the industry, Cincom was a mainframe player. That wasn't the only marketplace -- but Cincom was the only one of these who was actually supporting environments other than the IBM.

Everybody else was IBM only. When I say everybody else -- maybe somewhere else another company had similar adventures, but they were nominal.

We saw early on that the advantages in some of these other environments compared to the IBM environment was immense, and so we began to promote the environments we thought had the best chance. And the one we thought had the best chance to grow and develop was the new type environment being promoted by Digital. So we got solidly behind Digital and put a lot more energy into the Digital workplace. We moved our entire system software line there, and we began pushing and promoting Digital as another major adventure within Cincom. We eventually moved not only our system software products but our own manufacturing suite of application products over to run on the Digital line of computers -- and the customer preference for these shocked us.

DKA: What kind of time frame was this that you're talking about?

TMN: We got into Digital in the late seventies and were well down the road with Digital equipment in the early eighties. By the early eighties -- '82, '83, -- we were selling four out of every five of our manufacturing systems on IBM, and one out of five would be Digital. Now, we offered the same feature functionality, everything was exactly the same. We tried to promote to the customer their choice of Digital or IBM. It was completely their decision. It didn't matter. The boxes, the applications were identical, and so on, and we tried to work with both vendors, and we tried to be completely neutral on it.

By the late eighties, Digital choices were now four out of five, so they had moved from being one out of five for our manufacturing systems to four out of five. When we saw this happening, we knew that a radical shift had taken place, that Digital computers were no longer computers for the technicians, but were being used in a major commercial processing arena. And we knew if this could happen with pseudo mainframes, it would also happen in workstations and PC's -- that the economics and the preference for deployment and cost efficiency and flexibility of processing would prevail over centralized mainframe processing paradigms. So we did everything we could to begin our migration to a universal player who supported the entire processing paradigm of the customer -- which would be mainframes, midframes, and workstation -- all with exactly the same type of software so that a user could, in fact, deploy and implement and integrate all of these diverse processing platforms without concern to the manufacturer.

In effect, what we were saying to users is look, in the old days you would choose software based on whether or not it ran on your hardware, but now, you can choose hardware which runs on your software platform. And we began to show them that the software platform is where you make the large scale investments which endure over time. The hardware investment is obsolete every 18 months, so you shouldn't make decisions that revolve around hardware, whereas the software transcends the eight, ten, or twelve-year usage cycle. So we began to get the users to choose their software first, then add the hardware as appropriate. Now, of course, this was radically different from what was being promoted at the time, and it was also antagonistic to a lot of other messages.

So once again we found ourselves essentially that "voice crying in the wilderness." Many of the mainframe hardware, or software providers that I mentioned earlier poked fun at us and laughed at us because we were seeming to move away from the big iron, the mainframe environment. Some were quite outspoken proponents of IBM and mainframe only as being the way to go, and they weren't about to support these other environments. They were in effect trying to reposition Cincom as abandoning the mainframe.

In fact, we weren't abandoning the mainframe. We were in effect abandoning the idea of hardware being the center of focus, to a world where software was the center of focus. We promoted the idea that you should use software which would provide a platform to allow you to choose whichever kind of hardware you wanted. And so it was only an extension forward of the thinking processes we had begun to see in the early seventies. We had made the user independent of various mainframes, and now we were making them independent of any processing paradigm -- mainframe, midframe, or workstation. That is the strategy we began to promote in the late eighties and into the early nineties. I believe that even today we're probably the only software company that in fact promotes this as the proper approach.

Today you'll see that software vendors -- big, substantial ones -- are not even supporting a mainframe environment - and in fact saying that mainframes are dead, there's no place for them, buy workstations only. Typically, that's because they only have software that runs on workstations. But we believe that users' best interests are served by viewing themselves as a network, much like the telephone communications system. A user picks up the phone, and it may go out of his building into wire, but it may be then beamed up into satellite, it may go to fiber optics, or a microwave tower, it may go back into wire, it may be passed around in many different ways without the user knowing that it has gone through all these different communications media. He talks without concern. We felt that the user on a network with his keyboard should be just as insulated from the processing environment and all the software as someone using the telephone service is. Basically that is what we have been promoting, and continue to promote today.

It has some liabilities for us, because now we're promoting multiple environments and such a software development adventure minimizes the thrust we can have in any one specific marketplace. But we believe that this is the correct direction for the long term, and in time users will want software that will run across all of their environments identically. This is something I believe is the environment of the future.

DKA: Let me ask, going back, a historical question. As you began to do so much more with Digital, what was it like, or how different was it to work with Digital compared with people at IBM? Were they a radically different culture that you encountered, or were they relatively the same?

TMN: They were radically different cultures. Digital and IBM in those days, in the late seventies and early eighties, were radically different. That's because of their roots. IBM had grown out of the commercial arena and was primarily in the accounting fields and large scale commercial computing.

Digital had grown out of the technical area, and that's where it's roots were. Digital was moving into commercial computing possibilities with its computer and software, but the mindset and the thinking processes and the selling and the service and support, everything about Digital was still oriented around supporting the technical user, who was typically sold by a third party OEM or vendors and so forth.

Digital was never really in the realm of large scale account support teams, skillful sales organizations, or anything like that. Their thinking was, we will build a better box, we will build a better solution, we will have better technology, and discerning, knowledgeable users will recognize this and will buy us. IBM was of the mindset that good technology is important, but selling service support, account management, marketing tactics and so forth are far more important, and so you had these two different spectrums, and meeting in the middle somewhere, two diverse cultures. It was really sort of cultures colliding when these two organizations came into an account. One is promoting and pushing its better technology, the other is pushing and promoting good technology and excellent service and support.

DKA: It put somebody in the middle, like you.

TMN: Yes. We were trying to support both of them. And it was difficult because on the one hand we had to quite often prop up inadequate Digital selling efforts, and we had to do the account management and account marketing for them. It wasn't to our advantage there, and so the support of Digital which we had hoped to get in many cases became almost a burden or a liability. At the same time this was happening, IBM continued to view Cincom now as not only a promoter of alternate software, but in effect a marketing extension of Digital, Hewlett-Packard, and other organizations; so this made it even more difficult for us in the IBM world. So we got into an arena where we weren't able to properly position ourselves as well as some other companies have been able to do.

But those two organizations are radically different. I think one of the problems for Digital was actually their success and the growth. As they became a much bigger company, they began to try to become more like IBM, and try to make themselves into a "mini-IBM" and contend with IBM in IBM's arena. In doing so, they moved away from their strengths and played to IBM's strengths -- in this they were no match for IBM. The more they played IBM's game, the more difficult it became for Digital.

It is still a problem today. Digital, for example, as part of its downsizing process, has decided it will handle only 1,000 accounts itself, and all the rest of its accounts are going to be sold to third parties. This suggests that they realize they are a good technology provider, and they'll leave the selling and account support to third parties. IBM is doing some of that as well, but still IBM, deep in their corporate psyche, is account management and account control, and it's very important to them, even as they attempt to get more third party providers helping them.

DKA: Let me change the subject a little bit. To support your various ventures and your strategy of doing a number of different things, there was a strong need for research and development in your company, always. You mentioned in your early history how it was the same team that did the daytime selling of the services, who performed the nighttime R&D. Tell me how your R&D strategy has evolved in Cincom over the years as the company has grown and changed.

TMN: Well, we did do that in the beginning, but that was only to get some packaged products built. As we moved away from being a professional services company which also did packaged software, to a packaged products company, I thought through what our priorities should be. We wound up competing with IBM, and since I had worked so many years with IBM and had great respect for their marketing, sales, distribution, market presence, account management, and account control, I knew that there wasn't one chance in 10 million we could outsell them. So our only hope was to out-technology and out-service them.

We therefore moved heavily into becoming a big investor in research and development. At one time we had over 25 percent of our revenues targeted for research and development. This was at a time when two or three percent of research and development was typically being done in American industry -- perhaps a little bit more than that in software, but not much more. So we became a great believer in research and development. In the industry groups of which I was a part, I attempted to promulgate large-scale investment research and development as fundamental to our industry. We had to be willing to make those investments -- it was absolutely imperative that we do it or our technologies would come to an end of a life cycle, and our company would collapse with it. So we had to continue to make investments in-season and out, in good times and bad, so that you are always moving forward with new paradigms. That's been fundamental to Cincom -- large, large scale emphasis on research and development.

The flip side of that is that any company has only so much money to work with, and so we in effect deemphasized marketing and sales. Quite frankly, whilst this has given us a longevity of life because we've always had really truly excellent technology, at different times other companies who for a short period of time put more money into marketing and sales and promotion and advertising have leapt forward. The industry has become primarily a marketing and merchandising business today, and unless you're heavily marketing and merchandising, the audience doesn't know you're there.

Secondly, there are so many different technology choices that users can't analyze them all. They seem to go with what seems to be the most market-present, and so it's a very difficult thing for Cincom to find ways to continue to make the investment in research and development in order to have the premier technology, and at the same time have enough resources to gain more market presence. But we still believe that R&D is fundamental, and will continue to build, develop and deliver high quality products that are competitive over time. It's something we're loath to let go of.

One of the ways we're finding to solve this dilemma is to enter into research and development alliances. This is where we enter into development adventures with associated companies so that not only our own resources are working to develop products. This enables Cincom to put more and more of its total budget into marketing and merchandising.

Because our R&D has been increased over what we are able to do ourselves through the collaboration with many other allies, I believe this is a solid solution forward. Collaborative research and development is where we have perhaps 20, 40, or even 50 R&D partners who allow Cincom to market and distribute a composite of technologies which completely satisfy the total customer requirement. These technologies are all put together under a common architecture which we engineer, design, create ourselves.

So more and more in the future, Cincom will develop more and more of the central architecture, the central glue and system that holds everything together. We will subcontract out, or work with, other allies to put together the parts, pieces, and components into a synthesized whole, so that we can deliver, as Mercedes does, a top notch, fully integrated automobile, serviced and supported by the Mercedes company, but built by hundreds of different suppliers. This is the model we're taking forward and have been working over the last several years. In this way, we can bring better technology to the market than ever before, but also have the resources to expand our marketing distribution and our public awareness so that we can more fully participate in a world which has become one of marketing and merchandising.

DKA: Let me ask you one other question about R&D. To what extent are you continuing to work on problems that were established by your customer base, and to what extent are you working on problems that are set by where you think the future of the business is?

TMN: Okay, that's a good question, because we never think in a void. There's an old saying that whoever lives by the crystal ball usually becomes accustomed to a diet of ground glass. Ground glass doesn't digest very well, so we don't try to work with crystal balls. And even though some people like to say we're visionaries because we have had a lot of cases where we've been three to seven years ahead of the industry, the reason we were able to do this is that we work with leading edge customers. These customers have these advance requirements that no one else seems to worry about; while we say these are exactly the kind of things there's potential for. And if we conclude that there are a number of customers who say this is what they want, then we get behind it and we work with a federation of these leading edge customers who help to guide our thinking as to what they want and need and where they believe the world is going. We listen to top flight third-party gurus who either affirm, correct, or help guide our thinking, and then to this we bring our part of it-- the engineering, the development, the construction, and the improvement of the technology. We bring it back in to those particular customers to perfect the technology with their insights, and then later, through the leadership of these leading edge customers, we move the market forward.

At this stage, many other competitors are in the game, competitors who may be bigger, and better financed than we are, so we can't hope to hold 30 or 40 percent of the market. But we can garner a large percentage of the market in its inception stage. Then while we have a declining share of markets as they mature, we have an absolute growth overall while we're again looking at the next future. And the more rapid the dynamics change, the more rapid the paradigm shift, the more we have to be at this bleeding, leading edge working with leading customers -- so that's become a part of just our whole corporate psyche. That's the only way we know how to do things.

We have a technology group that analyzes and studies leading products from small companies who themselves have ideas about technology that should be developed. We might fund, and collaborate with them, or bring them into our particular development labs, or put some of our people into their labs. And so this federation, the pooling of knowledge, a sort of an advanced technology center that transcends an individual company --the virtual corporation, the virtual R&D organization, if you will -- is how we're doing things now. We're radically increasing our emphasis on R&D and leading edge technology at the same time we're expanding our marketing and distribution through an entirely different approach. Whereas before we did most of it ourselves, and plied it around the industry, today we're looking to do more and more based on what others are guiding us to do.

DKA: What you've talked about is certainly two strategies, companies usually take the one or the other, not take both at once. It must be a challenge.

TMN: Well, nothing's easy in this business. It's like trying to win the U.S. Open up in New York. I mean, some great tennis players will suffer some agonizing defeats there. It's top flight competition, it's tough, and it's not enough to know that you're one of the 10 or 20 best players in the world, because you have to win. And that's what we have to do. We have to win in the marketplace. We can't just create good technology for its own sake. We have to make sure it's the technology that will win. Sure, it's complex, it's difficult, and it's all-consuming. There has to be a willingness to sacrifice, a continuing willingness to sacrifice for the good of the technology, and for the advancement of the marketplace.

We constantly try to infuse into our people the old John F. Kennedy idea of -- ask not what Cincom can do for you, but what you can do for Cincom. The idea is that we want Cincom to do well for the user community, for the state of the industry, and for the advancement of the technology. We must be givers more than takers - even though we're in a world where most people are being taught to "look out for number one" -- so the idea of sacrifice and dedication and devotion to something that may not benefit you much personally is a concept that needs to be continuously promoted inside any organization. The organization itself has to believe it. Basically that is what we try to do to help ourselves, because this is a difficult, difficult issue.

DKA: Tom, you know, so many people who have followed a company from its origins have difficulty in dealing with a company that grows beyond their individual span of control. Has this been a difficult transition for you personally to help the company move to this next phase of its evolution?

TMN: Well, it's an interesting question. You know, I think the people who have difficulty in letting go, as you say, of an individualized span of control in some regards might be like the person who was a wonderful high school athlete, very gifted then, and he has never been able to grow beyond that. He always is behaving and thinking about the good old days in high school. He's unable to let go of the past, and then he's not able to participate in the future.

In looking at my own personal situation and career, it's a choice between trying to hold onto all the loose ends myself and attempting to integrate everything together, which I believe not only stunts myself but stunts the company. Worse, this sets a bad example for all the people I'm trying to develop as leaders and executives, so I believe such a strategy is harmful and in the end would be destructive to Cincom. So even though, quite frankly, I enjoy the involvement with all the areas -- I'm still actively interested -- I talk about the sacrifice that our people have to make, and I talk about putting the company ahead of personal interests, and the organizational goals ahead of individual goals, and I don't believe that I can in fact present that message with any sense of conviction or authority if I don't practice it myself. So the primary exemplar of sacrifice I believe must be made by the leader. The leader of every organization has privileges only because he has certain duties and responsibilities. If he doesn't put the duties and responsibilities first, he's ill-serving himself and his organization.

So making myself more and more dispensable to Cincom, I believe well-serves our organization, and is a major part of what I have to do. It's not so hard to let go of the past in some ways, because the idea is to get 25 or 30 years of experience in a career, not one year repeated 30 different times. And it's only by growing and developing and letting go of the past that I can grasp the future.

But it's a gradual process of continuously thinking, learning, and understanding how I can commit myself to the organization in a more productive way - both for myself and for the people who are dependent upon me for the proper leadership and guidance. Delegating, sharing of responsibility, sharing of authority, sharing of power, moving decision making into the organization, and then being willing to stand back and let people make mistakes and support them in spite of the mistakes is a major part, I believe, of growing for the senior executives and also for each of the other managers. The problem with this is that we don't want to make the same mistake over and over again. Becoming truly a learning organization throughout is what is essential to make an organization go forward.

You know, it's one thing to say, well, Cincom started in database, or we started in mainframes, and what about the future, and can we in effect survive in the future as well as we did in the past, and so on. I believe that ability is more a function of the understandings and the willpower and the determination of the organization than on its set of skills and know-how. For example, many people know that Honda came into America selling motor scooters and things like that, and today are one of the biggest sellers of automobiles in America. So they've evolved away from what they once did to something radically different. But most people don't know that Cadillac once made bicycles. They moved from a bicycle manufacturer to an automobile manufacturer -- but not just an automobile manufacturer, but the most prestigious automobile manufacturer. So when we say it's the Cadillac of its class, that says something about the niche of that particular product.

An organization evolves based on its ability to let go of its past, and an individual does the same, and so this constant process of studying and thinking and knowing where you're going is very, very important. When I was with IBM, Tom Watson, Sr. would give out placards, and one of the slogans they were famous for said simply: Think. You don't see those given out today, or you don't hear that talked about much, but I'm thinking quite frankly of doing that within Cincom, because the most important thing that our people can do is to think -- carefully, seriously, rigorously -- instead of just trying to act all the time.

Now, we do a good job, I believe, of thinking and acting, but too often the pressure to act and be busy prevails over the thinking process. Unless one emphasizes the thinking constantly, it sometimes takes second place -- and when we do that, we're giving up the most powerful resource that any human individual has, the power to think and reason. So we have to think our way forward as well as work our way forward, and we're trying to do that in our company.

DKA: You've mentioned that you talked about the growth, not only your own personal growth but the growth of Cincom, and that you need to not only look at what has made you successful, but look at some of the things that haven't worked, some of the mistakes that you've made. As you look back over your long history with Cincom, what are some of the things that you feel were mistakes in the past, or maybe missed opportunities that you've learned from, and what have you learned from them?

TMN: I think one of the major mistakes we made was that we never really bought into, as we should have, the importance of marketing and merchandising as an organization. We were more along the ideas of Digital, better technology will win. We believe deeply in technology, and we talk about marketing and merchandising and advertising and promotion as being important, but we don't really believe it to the same degree we believe in good technology, good service, and good support, and that's a mistake I think I've made.

I haven't properly led the organization in that regard, because of our roots. I knew we couldn't win and survive if we competed against IBM on sales, and I never was able to take us to the point where we were no longer competing with IBM, but were competing with a bunch of different competitors, and here marketing and advertising became much more important. So I believe that was a mistake. We're working hard toward correcting it, but still it's deep in the psyche of Cincom that if there's a dollar that has to be split between customer service and support, research and development for advanced technology, and sales and merchandising -- sales, marketing, and merchandising comes in third place every time -- and that's not good for our organization. We have to balance that better. That's a major problem for us.

A second mistake I believe we made is that we saw too clearly that there were multiple paradigms that had to be supported -- the mainframes, the midsize, and the workstations -- and we tried to support all three of them concurrently. And with the limited resources we had, we therefore weren't able to become a big player in any one of the three. We were always a supporting player in each of the markets we were in, rather than the dominant player in one. I now believe that, from a company gross standpoint, we could have grown much faster -- we saw workstations and all their potential six, seven, eight, or even nine years ago -- if we had completely moved away from our mainframe commitments and so forth, and participated fully in workstations and distributed processing -- in effect put the others just in a holding status. And so from a corporate standpoint one could point to that and say, that's a mistake. You made a mistake there. You should have really, for corporate growth and profitability, gone after that.

But we have 5,000 customers who had bought into us over the last 20 years. How do you in effect tell those customers that you can't really fulfill promises and commitments to them? Sometimes you make mistakes on the one hand so that you don't make them on another.

But in trying to support these multiple markets concurrently, we didn't orchestrate as well as we might have, and that's been a problem for us. So in effect, not being able to balance research and development, and customer service and support, with marketing and sales as well as we might have; as well as not being able to balance the three major paradigms (mainframe, midsize, and workstations) has been a problem for us.

Earlier, I used the analogy of the guy at the carnival show trying to keep all those plates twirling, and the guys at the carnival, quite frankly, are always able to do it. They never seem to come crashing down. But we have had our missteps along the way here and there, and it has not been easy to always keep everything working perfectly in such a terribly dynamic environment that we find ourselves in.

DKA: I guess one of the advantages, however, is that it gives you stability -- even though you may not have explosive growth, it may mean that high risk doesn't pay off. Was that part of your decision making? I mean, your growth has certainly been steady. It may not have been always explosive, but it's there. That may be a benefit of your strategy.

TMN: Well, that's a good point. There's a very, very good book out now called *Built to Last*, and it studies 18 major companies who are dominant in their field, and how they compare with their contenders. This book shows how the philosophy of those organizations were built to last, so that they go through peaks and valleys and so forth, but they're always progressing forward over the long term - and how everything about the company is always thinking about the longer-term horizons. That particular book does a wonderful job of showing exactly what we're talking about.

These ideas were always in my mind from the earliest days. We were building a company that was built to last over time. And this came from two primary ideas. The first one is directly from IBM. IBM is profiled as one of the 18 companies that this book said was built to last. Well, I was trained by IBM to think about the longer term. Maybe we don't have the best technology right now, but that's not important. We will have down the road, and we will do this and we will do that, so that this idea from IBM is a part of my being and is also part of the "being" of Cincom now. The second thing is the nature of the technology that we offer. We don't offer technology that the customer can buy, use for a while, and then throw away. He builds his applications around it. The company is completely and utterly dependent upon it. It is strategic software. It is the infrastructure around which the companies run. All of our software is that way.

It's like a person who is married with five or six kids. You're committed, and you don't leave that commitment very lightly without a lot of adverse implications -- and so everything about us has to be about stability, growth, security over time. Now, there have been in recent memory some meteoric successes, and we know many of the names of some of those companies, but if I were talking to you in 1980 we could talk about some of the meteoric successes of the companies in the late seventies, as well. And you might have said they've grown faster than Cincom -- but by 1985 or 1986, almost every one of those companies that you would have talked about would have gone out of business. They had meteoric rises, but then they plummeted almost as fast.

That's not to say that every great shooting star has to fall, but it is to say this: where is your priority? Are you willing to persevere over time, year after year, doing what you believe is right for yourselves, your customers, your people over the long term? If so, you are going to miss some of this explosive growth, but you're also not going to be feeling the pain of catastrophic failure when whatever it was you bet everything on eventually is no longer viable. And in a field of super high technology that is changing more rapidly every day, you can be sure that the life cycle of a technology emphasis will continue to contract, so that what used to have a life cycle of seven to eight years has now become three to four years -- and may soon be down to even a one or two year cycle. So it's very hard to bet everything on something like hula hoops, which have a huge craze for a year or two and then nobody buys them.

Stability over time, built to last, the commitments to the customers, the kind of business we're in causes us to be a company that is year-in, year-out trying to improve itself and better itself and grow. That's not to say that we're not looking for the explosive growth opportunities. We want to have them, but we don't want to be a hero today and a bum tomorrow. We want to take those and complement and build them onto what we're trying to do. There are some good opportunity areas for us today, for example, the area of object-oriented, the area of distributed and downsized manufacturing systems, the area of text and text application development. We've got four or five areas that look very, very good for us to have very big explosive possibilities in this second half of the 1990's decade and on into the 21st Century, and we hope to participate very fully in that.

And quite frankly, we've been making sacrifices over the last four or five years in marketplaces that may well be mature to declining markets so that we can participate in those markets of the future. But steady, reliable, predictable growth, security, is something that's very dear to us, and we want to have that.

DKA: Well, the point that you were just making about where the future lies is where I wanted to go as the final thing that we talk about. How do you see this area of software that you're involved in developing and changing over the next few years, or where will be the major opportunities? How can you position the company to make it so that it can realize those opportunities most effectively?

TMN: There is an interesting analogy presented by a very, very fine author from the East Coast that compares the idea of invention and innovation. In this example, he says there's a big difference between invention, which is where an idea is proved to work, and innovation, where everything is brought together that it can now be widely used and commercially viable. As an example, he uses airplanes and points out that in 1903 in North Carolina at Kitty Hawk, the Wright Brothers proved that airplanes heavier than air could fly.

Now, it doesn't matter that the plane only flew for 12 seconds and only 120 feet or so. The plane could fly. But it was another 35 years or so before air transportation became commercially viable, in other words, before it could economically fly as well as physically fly. In order for that to happen, a confluence of technologies had to come together into a composite which was known as the DC-3. Now, in the DC-3 what was brought together was a lightweight internal combustion engine, a body construction which was lightweight, variable pitch propellers, a retractable landing gear, and wing flaps.

The plane needed all of those things to become commercially viable, and really solve the customer's problem. Until the DC-3, air travel was not economically viable.

Now what has been going on over the past 10, 12, or 14 years in the computer industry is similar. We have a variety of technological components which need to be integrated into a technology ensemble - an ensemble so that within one technology, you have all the solutions enabling problems to be solved quickly and easily. Then you will have a tremendous breakthrough. Until that happens, the users have to pick and choose diverse technologies and fit and weld them together -- as though they're buying their own engine, their own landing gear, and so on, and trying to service their own airplane. This is really not viable. Even though people are selling billions of dollars of software a part at a time, it's not really what the customers need and will demand five to ten years from now.

Where we believe the future lies is in a technology ensemble which brings together into one system an integrated set of technologies which automates the process of building, developing and maintaining applications so that we have orders of magnitude improvement over current environments. I've set for our company the challenge of "200 to 1 by the year 2001". This means that by the year 2001 we will have increased the capability of building, developing, and maintaining applications by a factor of 200 to 1 over what it is today.

This integration of diverse technologies, we believe will be and must be built around object-oriented technology. Object-oriented technology will enable end users to construct their own applications without building, developing, and maintaining them. The software in effect automates the building and developing processes. As computers are designed to automate processes, we need to automate the process of building and developing applications so that we can have a 100 or 200-fold implementation of computers and move them rapidly everywhere.

In order to make this happen, we need new technologies in the area of object-oriented development and object-oriented database management. We need new technologies not only in those areas, but also in the areas of automated workflow so that work doesn't move physically from desk to desk, but rather it's all moving around networks. In this way, people are integrated into automated work flow, and what moves around those networks are objects. The world of the future will be object-oriented.

This means that companies that are down the road with relational, down the road with some of the conventional application development technologies, who are not developing automated workflow around these new processing cycles, will be very, very severely handicapped when their competitors can do in a day what takes them two, three, or four months to do. They will be able to cut their cost of developing end products or services by factors of 20, 30, or even 40 percent, because the information flow content is dramatically reduced.

We want to use automated technology and computers and software to change the way the world thinks, works, services and supports its customers. We're on the very beginning edge of a major, radical revolution which will mean as much to the 21st Century as the industrial revolution meant to the 20th Century.

In the late 19th Century, when the industrial revolution developed, one 140 pound worker could sit on a tractor and in a single afternoon, plow as much field as 200 Arnold Schwarzenegers could previously, pulling plows by themselves. Today, we're on the verge of a similar breakthrough which will enable one end user with the right kind of software technology may be able to build, in a single afternoon, an application that it might take 20, 30, or 40 developers a month or two or even several months to build.

The tools and technologies have already been invented. They haven't yet been innovated and brought into an integrated set of technologies that satisfies the ongoing requirements for customers. And we believe it's the companies like ourselves who see and understand this and are willing to make this available across a broad array of processing paradigms - who are willing to forsake current market opportunities and invest in R&D and training and teaching, who will be major participants in the 21st Century. We also believe it's absolutely imperative that this be done.

DKA: Could we explore a bit more how this automation and applications you were talking about in comparison with the industrial revolution of the 19th Century and the early 20th Century can have a similar affect on the information industry, particularly the way software has become the central driving force.

TMN: All right. Well, as I mentioned before, it's necessary that we make this radical revolution in the way we build, develop, and maintain software so that we can really, universally use computers on a broad scale, and in that process, cut the cost of building, developing and maintaining applications by orders of magnitude. I mean 200 to 1 over what they are today -- very, very big, not just marginal improvements - radical improvements. And it is necessary to employ these things as quickly as we can in order to compete.

We're in a global economy today, and cost of labor and the ability to come up with new products is such that companies which are no longer competitive can be devastated very quickly. We've seen this happen to General Motors in the 1980's. We saw IBM and Digital go down so fast, I mean, it was almost unbelievable what happened to them. IBM in the space of a couple of years lost total market share value of its company over \$100 billion dollars. Now, to put that in perspective, that's more market share value than the most highly valued company today, or in the range of what a Standard Oil would be valued at -- lost in the space of a couple of years time because the company was no longer competitive. The investors saw it, they didn't like the results, they dumped the shares. With Digital, we saw the same kind of devastation, the stock going from \$200 to less than \$20 a share, a 90 percent loss in its capital values.

Now, if it can happen to companies with that might and power, then certainly it can happen to the \$200 to \$300 million companies. Certainly it can happen to the \$5 billion company, unless they are really on top of the technology, because it's the technology that will make the difference. And the technology that will make the difference is software, not hardware. Software gives life to the equipment. Software is the solution. In the past we talked about the software running on the hardware as though the hardware were the investment and the software was the expense.

In fact, the software is the investment and the hardware is the expense. The software is where the risk of failure is. The software is where we have to put most of our money, time, and interest, and so on, so that in time the industry will become a software industry, with hardware providers supplying the technology that runs on those software platforms. I wouldn't be the least bit surprised in years ahead to see IBM becoming primarily a software company, which may mean subcontracting out its hardware manufacturing around the world. In fact, I'm not crystal-balling right now. Today, their PC's are provided them by providers all over the place. When you buy a Hewlett-Packard computer, you've got Intel inside. Very few computer manufacturers today are manufacturing computers. They're assembling computers.

The solution to the customer problem lies in the software. The money, the profit, the potential is in the software. So we've come full circle from the '60s when IBM used to tell me the money's in the hardware, the problem's in the software. In the 21st Century, the money and the opportunity will be in the software. Smart company executives will see this and will rapidly transform their companies into being solution providers, which will include software in its diverse forms -- packaged software products, professional service support, advice and guidance, maybe even processing services much as EDS does. There will be a radical transformation into what really goes into making a solution -- the hardware that may be provided could come from many different companies.

It wouldn't surprise me, and it doesn't surprise me today to hear IBM and others saying, look, we're interested in providing a solution. It may or may not include IBM equipment, because this is what's going to happen. You don't need to gaze into the crystal ball to predict what's going to happen. All you need do is look into what is already happening and project it on a broader basis -- and all of the things that we're talking about today are already happening. They're in the system. They're being implemented. The smarter companies who are in the business of merchandising are doing this, and the smarter customers are buying into these ideas -- so it will be a radical transformation.

Some years ago I wrote to Cincom'ers that in the future IBM may change its name from IBM, meaning International Business Machines, to IBS, meaning International Business Solutions, or International Business Software. It was just a joke we talked about, but you see the transformation already happening, and it will continue in that direction. Customers want solutions. There's no magic in the boxes any more. The days of the beautiful computers with flashing lights and consoles, and all that we all fell in love with is just not there any more. Solutions are what customers want. Solutions are what customers need. Solutions are what people are willing to pay for, and providers always go where the money is, so in my opinion that's where the future of the business is.

DKA: Thanks very much. This has been terrific. We've really enjoyed hearing your perspective. It's great to hear it from somebody who's seen so much of it.

TMN: Thanks. Thanks a lot.