

An Interview with

MARTIN GOETZ

OH 334

Conducted by Jeffrey R. Yost

on

3 May 2002

Washington, DC

Charles Babbage Institute  
Center for the History of Information Processing  
University of Minnesota, Minneapolis  
Copyright, Charles Babbage Institute

---

## Martin Goetz Interview

3 May 2002

### **Abstract**

Martin A. Goetz, a software industry pioneer, was a founder and past president of Applied Data Research (ADR). He was awarded the first software patent in 1968 for his sorting system program and was a longtime spokesperson for protecting software as intellectual property. In this interview Goetz discusses aspects of his early career as a programmer at Remington Rand, his founding of ADR, his management of ADR's Software Division, ADR's competition and litigation against IBM, IBM's unbundling, his work with ADAPSO, and his leadership on the issue of the intellectual protection of software. He concludes with some remarks on the relationship between anti-trust issues with IBM in 1960s and 1970s, and the Justice Department's case against Microsoft this past decade.

---

## Preface

As part of its preservation activities, the Software History Center (SHC) worked with Dr. David Allison of the Smithsonian Institution's National Museum of American History and Dr. Jeffrey Yost of the Charles Babbage Institute to plan and conduct a number of oral history interviews of early software company founders and other key industry contributors. On May 3, 2002, in conjunction with SHC's ADAPSO Reunion meeting held in Washington, DC, SHC arranged for 15 individual interviews by historians well qualified by their knowledge and interest in computing history.

The following people were interviewed together with the name of their interviewer:

Bruce Coleman, interviewed by William Aspray  
Richard Crandall, interviewed by Paul Ceruzzi  
Gary Durbin, interviewed by Philip Frana  
Martin Goetz, interviewed by Jeffrey R. Yost  
Bernard Goldstein, interviewed by David Allison  
John Keane, interviewed by Martin Campbell-Kelly  
Ernest E. Keet, interviewed by Philip Frana  
Frank Lautenberg, interviewed by Paul Ceruzzi  
John Maguire, interviewed by William Aspray  
Joseph Piscopo, interviewed by Thomas Haigh  
Lawrence Schoenberg, interviewed by Martin Campbell-Kelly  
Charles Wang, interviewed by David Allison  
Robert E. Weissman, interviewed by Paul Ceruzzi  
Lawrence Welke, interviewed by Thomas Haigh.  
Sam Wyly, interviewed by David Allison

Each interview was tape recorded, transcribed and edited by SHC, the interviewer and the interviewee to ensure clarity and readability without changing style or flow. The original tapes along with the edited transcripts were donated to CBI, which placed the edited transcripts on the CBI website.

---

---

## ADAPSO History Program Martin Goetz Interview

---

**Jeffrey Yost:** I'm here today with Martin Goetz. It's May 3<sup>rd</sup>, 2002 and we're in Washington, DC as part of the ADAPSO Reunion. This oral history is being sponsored in part by the Software History Center, the Smithsonian National Museum of American History, and the Charles Babbage Institute. These questions were developed to complement Martin Goetz' memoirs in the IEEE Annals of the History of Computing (Part I V.24:1; Part II V.24.4, forthcoming). Martin you joined Remington Rand in 1954 after studying business at City College of New York. What led you to answer an ad for a programmer or a trainee and at what point did you realize that you had a real affinity for this area?

### **BACKGROUND**

**Martin Goetz:** Well Jeff, I actually was looking for a job. I had gotten out of college and went to work for an advertising agency doing statistical analysis of Nielsen Ratings, which at that time was for radio only. And I wasn't happy there. So I answered an ad in the *New York Times*, and they said no experience required. They may have said college graduate, but as it turned out most of the people that were hired weren't college graduates. They came from a whole variety of backgrounds. In fact as I recall, very few had mathematical backgrounds. I had majored in business statistics. I think they were looking more for young bright people with logical backgrounds and logical minds but there was no test given. It was purely on the basis of interviews, and we went through a fairly long training course – 12 weeks, but other than that, there was no particular requirement for programming.

### **SPERRY RAND**

**Yost:** During the second half of the 1950s Remington Rand and Sperry Rand after the merger in 1955 had a difficult time competing against IBM. How much of that do you think was missed opportunities in marketing or managerial difficulties or was there much of an opportunity given IBM's account control?

**Goetz:** Well I think Sperry Rand missed a lot of opportunities, but as it turned out both IBM and Sperry didn't really realize the potential of computers within business organizations. Prior to when I went into the field there were just a couple of commercial installations, mostly insurance companies. But most of the other systems that were sold were sold into the government, so Sperry certainly didn't see the potential, but IBM actually didn't see the potential either. In fact they first predicted, there was a smaller computer called the IBM 650, and they thought they might sell several dozen, and they sold thousands of these smaller computers. As far as account control, IBM did have account control through their sales of tabulating equipment and they basically had about 80-90 percent of the punched card field when I started. Sperry Rand's cards had round holes which probably are antiques now. So they were also in the punched card field, but they had a very small share of the market. So I think IBM account control certainly helped them get business for the emerging data processing field.

**Yost:** Did you and all of the other people who founded ADR in 1959, have experience with Remington Rand?

**Goetz:** Yes, all of us. Some had worked directly for Sperry Rand, and those are people that I had known; others came from the C & O Railroad which was a large Midwest railroad that had Univac equipment. So we all came together in Princeton, New Jersey, where one of the founders, Woody Kaufman, had lived. But we all had Sperry Rand or Univac experience.

**Yost:** Given Remington Rand and Sperry Rand's difficulty in competing against IBM, were there any particular business or technical strategies that this group had to more effectively compete against IBM, or did you see yourself as staying away from IBM's areas?

**Goetz:** Well definitely when we got into the software products business we wanted to stay away from IBM. The first five or six years we did programming services and at that point we competed with other very early software companies. In particular, Computer Science Corporation was formed about the same year ADR was formed. There was a company called Computer Applications and we competed against them. So it wasn't until we got into the software products area that we competed with IBM, and then in a certain sense, indirectly.

## **SORTING**

**Yost:** Can you describe how you personally got involved in programming sort systems?

**Goetz:** Yes. My first assignment after finishing training was to go to work for Con Edison. At Con Edison they were building a billing system for gas and electricity customers and in the design there was a lot of sorting that was required, and in particular there was a requirement to sort ten word records which were 120 character records. And the only sort system available for that was from a sort generator which a famous person, Betty Holberton had built and donated to the Univac library. In using that system it turned out that it did the job, but it was slow. And I just took it upon myself to try to write a ten word sort and I did write it. It was significantly faster because it was specifically for ten word records. Univac was a word machine. So I just sort of fell into it. I really had never thought about sorting until there was a need for it within Con Edison. We were in direct competition with a similar application being built by another group within Con Edison that had IBM equipment. Sperry Rand was in there basically competing against IBM to see which company would survive within Con Edison. So, not just having an application built, but having it run fast was very important and that was one of the reasons they asked me to look at the sorting needs within the Con Edison application.

## **ADR MANAGEMENT**

**Yost:** In the 1960s there was some turnover and managerial changes at ADR and you've remarked that Dick Jones, former Univac sales manager, took over as president, and the firm began to grow much faster. What did he do differently?

**Goetz:** He was a very marketing-sales oriented person and I think he was much more a people person. Woody Kaufman was very technical, very bright, he used to still program

even though he was president of ADR. He'd come in at usually twelve or one o'clock in the afternoon, after working all night programming and I don't think he really was meant to be president of the company. He joined as the first employee. He was a terrific guy, but some people are best not being the president. I didn't want to be the president for many many years until the 80s. So we got a marketing-oriented person. Dick Jones was a good leader. He was a good salesman. Not necessarily a good manager – certainly never thought about cost controls and budgets which was something that as ADR grew became much more important than it was in the early days.

**Yost:** Later on you became very involved and played a important leadership role in intellectual property issues which we can discuss in the latter part of the interview, but did you ever regret the firm being as open as it was with its technology in the first half decade to two decades, such as participating in the IBM Sort Symposium or was this just good business and publicity for the company?

**Goetz:** You mean the ACM Sort Symposium? We were getting a lot of sorting business as it turned out in the early 1960s. I don't consider ADR being overly open in terms of its technology. At that time we were in the programming services business. We didn't think about protecting any of our intellectual property until we got into the software products business about 1965 when I met Mort Jacobs, who was a patent attorney, at a conference. We had sponsored a previous ACM symposium back about 1962. So we had done it once before, and sorting was an important area when most computers had only tape systems and didn't have disk storage. And ADR was into that area so it just seemed like a logical thing to sponsor the symposium. At that point we had not filed for any patents. It was a way to promote ADR, get its name known.

## AUTOFLOW

**Yost:** Can you briefly describe the origin of AUTOFLOW?

**Goetz:** Yes, AUTOFLOW happened really just by chance because companies building applications were requiring flow charts, but many programmers never flow-charted. It just became sort of a requirement, a documentation requirement and RCA, which was in the computer field at that time was looking to build some software to give away to its users and thought an automatic flow charting system might be of interest to their users, to help sell their hardware. And they came to ADR and asked if we might be interested in building such a system and we said we were. We gave them a preliminary design with a price for what we thought might be new business. As it turned out, RCA decided not to build such a system and at that time we had a design and decided we'd build a prototype system and maybe sell it to other hardware manufacturers. When we built the prototype we built it on an RCA 501. We went to other manufacturers who didn't really have any interest in funding such a project. Having built this prototype system we decided we'd try to sell it to the RCA user base, which only had 100 RCA 501's. And that's how we fell into the software products business. We really didn't think about software products per se, but we ended up building one program. That was an era when a manufacturer would sell their hardware and have maybe ten or fifteen programs, usually programming tools, that they would give to their users, and that was about the extent of the programs that were available to the user. So it

was proposed to RCA to expand their 10 or 12 programs. We thought they might give the user a program for their documentation. That's how it all started.

**Yost:** At the time of trying to sell it to firms with RCA mainframes, or subsequently when a version was developed for the IBM 1401, did you see the company as consciously entering the software products business, or was it a real redefinition of the company?

**Goetz:** No. It really wasn't a redefinition. I was always interested in programming tools at Sperry Rand and at different installations where I worked, even prior to going to IBM, which I went to for about one year. I was always interested in programming tools. And the flow chart system that we built, AUTOFLOW, was just one example of a programming tool which, as it turned out, was very useful during program testing because it would give you a logical view of your program with lots of cross-references. It actually helped during the debugging phase. But as we saw that there was perhaps a need for AUTOFLOW and started concentrating in that area, still most of our income came from programming services, and I'd say in 1964 and 1965 we didn't quite have the view that there was a software products business. It was more like 1967-68 that we recognized, yes there is really a business there and some other software companies had software products, so we weren't quite alone. I'd say that in the 1964-1965 period we didn't quite recognize the potential, but shortly thereafter we did.

## **SOFTWARE PRODUCTS**

**Yost:** Later in the 1960s when you began to realize this potential how did marketing and sales for products change from services? Were there different methods or techniques?

**Goetz:** Yes. When we started out in 1965 we had three offices. One on LA and one in Washington and I think we may have had one in New York, but that may have been a little bit later. But the people that were selling programming services were also trying to sell AUTOFLOW to prospects, which were basically 1401 users. And it turned out that it really was a long selling cycle and you had to concentrate to understand AUTOFLOW and understand the user needs. So slowly we evolved and started to get full-time sales people. And recognize that we really had to have people who would understand the product and understand the needs of the users. So we evolved from having part-time sales people to full time sales people selling software products exclusively.

**Yost:** Was there a geographic concentration of where you were selling or was it nationally?

**Goetz:** We were selling nationally, and we would fly our salesman to the site. As it turned out AUTOFLOW was a very nice system to demonstrate. This was not the first version; this was after you could just take a program, as is, and run it. We would take a user's program that we had never seen, run it through AUTOFLOW at his site, show him a flow chart on the printer, show him parts of his program where he could never enter, places where the system would have no way to go, that was just extraneous code. So it did a logical analysis of the program prior to printing the flow chart. It turned out to be a very good debugging tool; it was a terrific program to demonstrate because you'd see a nice two-dimensional flow chart

coming out of the printer within about five minutes of putting the cards into the card reader. You still needed punched cards when we were selling AUTOFLOW.

**Yost:** And what were the primary types of firms you were selling to?

**Goetz:** Very large firms. I remember Bordens being an early firm and Shell Oil. In fact I sold it to Max Hopper at Shell, who became a well-known manager at American Airlines and other places; he was a very early customer. So there were very large Fortune 100 customers. NASA turned out to be a large government user very early on.

**Yost:** ADR also got into time-sharing service bureaus and turnkey systems. Were these businesses profitable for the firm?

**Goetz:** For the most part, no. We were feeling our way in the 1960s. Time-sharing started to become popular and John Bennett had an interest in that. We got into the Service Bureau business. We ended up acquiring a company in 1967, Massachusetts Computer Associates that basically did research for the government. John Bennett was at that time in Washington as our branch manager so he was selling programming services and decided to get into the service bureau business which was reasonably popular at that time. It was in the late 1960s that we got into the time-sharing business. But I moved away from programming services and started concentrating in 1966 and 1967 purely on software products because we decided we were going to start building some other products, having reasonable success with our first product.

**Yost:** Did you feel that it would have been better to concentrate more of the firm's resources on software products as opposed to these other areas?

**Goetz:** Well, I don't think I really thought about it at the time. I was happy that Dick Jones gave me the freedom to work on some additional software products. Dick was very excited about the potential of software products. In fact he commissioned people in Washington and Massachusetts to build software products. I had enough on my hands building some new products in Princeton and was very happy doing it. I didn't have responsibility for profit and loss although I was a director of the company. Having this freedom from Dick Jones was very important. He basically gave me the freedom to hire people to build some new products, and at that point I really wasn't thinking too much about these other areas but was concentrating in the area that I was really involved in.

## **NEW SOFTWARE PRODUCTS**

**Yost:** Can you discuss some of the major new products that were developed?

**Goetz:** One was the Librarian, which turned out to be a very popular product, which helped you maintain your source programs, and keep them secure. That was a product that we built for internal use. We were basically an IBM shop and our programs were on punched cards and that was not a place we really wanted to maintain systems. As it turned out, when I started with Sperry Rand we were actually using IBM punched cards. They had a punched card reader as input because most of their users had lots of punched cards. When you wrote your programs you generally would write them on a system called the Unityper



which used a keyboard and your programs were stored on a little mini tape. So I was used to working with programs that were stored on tapes and not on cards and as the AUTOFLOW system began getting big we built the Librarian for in-house use. We recognized that it was something that customers might be interested in, so initially we offered the Librarian as an option to AUTOFLOW, in the very early days. And then we decided to separate it into being its own product and so we had AUTOFLOW and the Librarian.

We also had gone public in 1965, so we had some additional capital to invest and in 1968 we had a secondary offering and that gave us more money to invest in new products. So in the 1967-1968 period we were looking for programming tools and became aware of a system that RCA had actually built. It was a turnkey system as it turned out; it was a hardware-software system for COBOL online debugging. It was called the Octopus. It was a Spectra 70 hardware system with lots of terminals. You could picture an octopus, and it was a system that would be dedicated for programmers for doing online testing. And we thought of building a system like that for the IBM 360 series, which had come out in 1964 but by 1968 was really the major system that IBM was selling. So we built a system that would emulate what these RCA Spectra systems would be doing. The concept was that it would be online COBOL debugging and that was basically a system that ADR originally called Conversational COBOL. But we later broke it into two products, the first of which we named Roscoe and the second we named MetaCobol, but its origins were a system called Conversational COBOL that we had built for the IBM 360 series.

**Yost:** With the initial public offering in 1965 you raised over six million dollars, yet at that time were bringing in less than a million dollars each year in revenue. Were you surprised you were able to raise this much outside capital or was that what you expected given the environment?

**Goetz:** I really don't remember the specifics. I remember we were going to go public. We weren't desperate for money but our cash flow was not great. We were living from hand to mouth. I don't remember it exactly being six million. It was in that neighborhood. It was money to be used for expansion and to increase our marketing and our development. We used it for developing proprietary software. But looking back, it seemed like a lot of money given that we only had a million in revenues, but I wouldn't call it a surprise at that point in time since there were other companies going public. It certainly wasn't like the Internet era, 1999 and the year 2000, but it does seem like a lot of money to raise given the size of the company.

**Yost:** Can you talk about the development costs of Roscoe and the Librarian and MetaCobol and the revenue they produced compared to AUTOFLOW?

**Goetz:** Well Roscoe produced very little revenues for many years. I think we actually made our first sale about 1969 but there were very few sales until about 1975. The Librarian on the other hand took off very quickly. It was very well priced. Its practicality and its functionality were obvious. Companies were not happy maintaining card decks and if you replace a card with another card there is absolutely no way of keeping track of it; so from an auditing perspective, forgetting about security, there was no way to maintain an audit trail of changes you made to a program. It was just sort of an obvious type of application, so we

sold lots of systems. They were low priced and sold very quickly. It was always very profitable for ADR.

**Yost:** You've written that by 1970, or the early 1970s software products were half the business. What was it roughly in 1965 or 1968? At what point did it grow to become half?

**Goetz:** I don't know exactly. I covered it in my *IEEE Annals* Memoirs. I think by about 1975 it might have become about 50% of our revenues – it continued to go up until in the 1980s it was up to 80-90-95% and basically became our main business. But it started off slowly. I'd say in 1970 it might have been a third of our business and then slowly moved up to about a half of our business by 1975, and we slowly sold off some of the other businesses. We got out of the time-sharing business. That's probably the only one area that we got out of, but we didn't grow the other areas. There wasn't as much opportunity. So it was really just the growth within the software products business and by the 1980s we were known purely as a software products company.

### **IBM UNBUNDLING**

**Yost:** Can you briefly characterize the public relations battle with IBM in the second half of the 1960s?

**Goetz:** Yes, I'd say in terms of the issues that ADR was involved in, primarily getting IBM to unbundle, IBM was pretty quiet. In other words there was lots being written in 1967-68, certainly in 1969. By that time, January of 1969, the government had sued IBM, not only for bundling which was just one of several areas in the anti-trust suit. About 1966, 1967, as we were competing with IBM, we started becoming more and more vocal about how difficult it was to sell not just AUTOFLOW against free software -- IBM had a free flowcharter -- but against products that we were planning on building. Our strategy was to build products that IBM did not offer free. We had no idea, no way of knowing what IBM was planning, so as a public company it was prudent for us to make our position known to try to get IBM to unbundle. And there were articles in trade publications about unbundling. Software companies were being formed in 1967 and 1968, so the environment for unbundling was starting to build up, and IBM was just very silent. As it turned out, as covered in the *IEEE Annals of the History of Computing* starting about 1965-1966, IBM internally was looking at unbundling --- whether it would be feasible from a competitive point of view and from a business perspective. So IBM internally was looking closely at unbundling, but publicly they were very quiet. We knew we were on the right side because it was obvious that it was hard to compete against something that was free. So it was good PR, it got our name known. It was also important for our growth and as a public company to push hard for unbundling.

**Yost:** I should mention that your Memoirs, the first part was published in *IEEE Annals History of Computing* in volume 24, number 1, January-March 2002 issue and the second part will be published – has the issue been determined?

**Goetz:** Yes, it's planned for the fourth quarter of 2002.

### **ADR VALUATION**

**Yost:** When the market valuation of ADR reached \$40 million in 1969 and the company was doing roughly \$6 million in sales, was there a sense in your mind or others at ADR that this was irrationally high?

**Goetz:** It seemed high. Some of us did sell. I sold a very small percentage of my stock in 1968 when we had a secondary offering, but until we got acquired late in 1986 I never sold any more stock. It was really all paper profits and we didn't think too much of it. It seemed high at the time but other stocks were high, a little bit like the Internet craze; it was unbelievably high but we had nothing to compare it with so it was there, paper profits. I never lost any sleep when it went down because I had no plans to sell it. But it did go unbelievably high or irrationally high. When it started going down in the early 1970s, it went to an unbelievably low price and then slowly recovered in the late 1970s.

**Yost:** Given your experience with this did you anticipate the Internet bubble?

**Goetz:** No, but I had seen other bubbles; there was an electronics bubble of technology companies, not software companies per se, I'd say in the late 1960s early 1970s, so there have been other bubbles. I became an investor in the early 1990s and did some investing in Internet companies, not a great deal, and generally was very careful in terms of investing, but the valuations were unbelievably high for the Internet companies – they never became that high for software companies. But I think, over time, logic prevailed and they came down to realistic values.

## **SOFTWARE PATENTS**

**Yost:** Can you reflect on what it meant for you personally as well as for ADR when you received the first software patent?

**Goetz:** I certainly was elated. By 1968 I had been involved in arguing about the patentability of software for about 3 years, so I really believed that it was patentable subject matter. I knew that at some point in time the patent office would recognize it because back in 1965 they had provided guidelines for patenting software. IBM on the other hand was very much against the patenting of software. There was a presidential commission about 1965 that IBM was very involved in. IBM was fighting against the patenting of software and ADR was for it. I really believed that it was patentable subject matter. I mean there was no reason why software should be excluded. Certainly it was not mathematics. We were building systems, and could have been building hardware systems, and certainly hardware was patentable subject matter for many many years. It was patentable subject matter. We thought there was no reason why software should be excluded. And also for marketing reasons we were feeling like second-class citizens. Here software was free, here software was not patentable. So it was all part of our strategy, our direction of elevating the view of what software was. It shouldn't be free, it should be patentable, it was all part of our story. And we were very happy and the first patent certainly was not a trivial patent. It was not an obvious type of an invention, so we felt very good that we finally did get the recognition.

**Yost:** Was there a sense at that time that a great change had occurred?

**Goetz:** I think it was all another step in recognizing that software was a business - that software was something that should be sold. IBM was still promoting that software was a service. They were putting all of their software in the public domain. Some people believe still that software should be in the public domain. But I'd say it's a very very small minority. Software companies don't differ from any other company - it's a profit-making organization and has to protect its assets and patenting your intellectual property is one way of protecting your assets.

**Yost:** Was IBM trying to fight against the patentability of software using the "software as mathematics" argument?

**Goetz:** IBM wasn't trying to use that argument. I would say their argument was software was different, and they basically wanted to come up with a registration system, somewhere between copyright and patentability. So it wasn't that they said there should be no protection, they just didn't think that the patent system was the right type of protection. Of course they recognized later on that we're not trying to patent the program, we're trying to patent the process. If anything, you're trying to protect the program through the copyright system, which has nothing to do with an inventive process. So they were against the patenting process, but their logic was not that it was a mathematical process, just that it was different. They came up with a registration system, but apparently they could not convince the world that it was needed.

## **PROGRAMMATICS**

**Yost:** Was Programmatic's ADR's first major software acquisition and why did ADR president Dick Jones change the model from internal growth to an acquisition strategy?

**Goetz:** Programmatic's was not the first acquisition. We acquired a company in about 1965 called Massachusetts Computer Associates that did research for the government and I don't think Jones had a model. I think his goal was to just grow the company wherever he could see an opportunity. We had worked with Massachusetts Computer Associates on a joint contract. We got to know the people, he liked the people, so in 1965 he proposed that we acquire them. They weren't losing money. They were making some money in government contracts. It was not a big acquisition. Programmatic's, on the other hand, was on the west coast in Los Angeles. They were in the programming services business, but they had built a sort and they were trying to market it against IBM's free sort. Dick Jones was enamored with software products. He was enamored with the CEO of the company who he thought very highly of, so he pushed for the acquisition which I was very much against. John Bennett, who was director of marketing at the time, and had moved from Washington to Princeton, was also not in favor of it. To compete against a free sort was not something that we wanted to do. Nevertheless, we did acquire the company because Dick Jones pushed very hard for that to happen.

**Yost:** Was it also a sense of an added burden on the sales force that there were too many products that they had to sell?

**Goetz:** Yes, I thought so. I was not, at that point in time, directly responsible for the sales force. John Bennett had come in as director of marketing under Dick Jones in about 1968

and at that time moved from Washington to Princeton. So it was more his responsibility and not mine. I was very much involved in trying to get our new software products out. I was involved with fighting IBM, getting AUTOFLOW to be successful. So I had my say and was unsuccessful in convincing Dick not to acquire Programmatics and went on with my business.

## **AISC**

**Yost:** In 1968 a software trade organization called AISC was formed. Can you speak a little bit about that and ADR's involvement?

**Goetz:** ADR and Dick Jones and I in particular were both very involved in fighting IBM, trying to get IBM to unbundle and trying to get patenting of software. This early trade association was formed. Dick Jones was the ADR representative and I think at one point, either 1969 or 1970, was president of that group. So I didn't get involved until about 1970, after Dick Jones left ADR. But it was really the first software association to be formed and there were about eight companies that were part of the formation. I had little to do with the group until 1970, but they were certainly supportive of ADR's position on unbundling. I think there were several software companies that were in the initial group. One was Informatics. There were some programming services companies in the group and there were some companies that did business primarily with the government and were involved with some other issues. For instance, one of the companies was very concerned about the Rand Corporation, which was a non-profit company doing business with the government, and perhaps undercutting prices because they didn't have a profit motive. So there were other issues besides unbundling, but the patenting of software and unbundling were two of the issues that the AISC supported.

**Yost:** Do you see IBM's anti-competitive practices as a fundamental emphasis for AISC forming in 1968?

**Goetz:** I don't know if that was the fundamental reason. I just think it's a natural thing for companies that have common problems, to want to form associations. So when they were formed I think it was also for the recognition of software services. I'd say there were more software services companies at that point in time than software product companies. I think they just found it was important to share problems.

## **COMPETING AGAINST IBM**

**Yost:** In 1971 ADAPSO started a software section and the following year AISC merged with this division. You've described this as the true beginning of the software products industry. Was this the first time that many companies thought that they might have a chance of competing against IBM?

**Goetz:** I think all the companies that were formed thought they had a good chance. There were many companies that were formed in 1969 and 1970, when IBM announced their unbundling – they announced it in June of 1969 and actually unbundled in 1970. So, many companies were formed and were very optimistic that this was a very good opportunity. But I think when AISC merged with ADAPSO it was really the first time that we had a real

representative group of software product companies that could share their problems. But there weren't too many companies at that time directly competing against IBM because even though IBM unbundled in 1970, all of the existing programs that they had at that time were still being given away free. And they didn't have that many programs, so for the most part, software product companies were building software that IBM didn't have. There was one company called Turnkey Systems International that had a product that competed against IBM's CICS. In the early 1970s there were some companies that started competing against IBM's database system, IMS. But in the very early 1970s most of the companies were not directly competing against IBM.

ADR was probably the exception. We were directly competing, but not so much with AUTOFLOW, because by that time the users were recognizing that IBM's Flowcharter was really a manual system. The users described each page of the flow chart and then basically key punched how they wanted the flow chart to look. There was very little that was automatic. It was becoming less of a competitor in the early 1970s.

We had built this system called Roscoe and IBM was giving away their product called TSO. It was part of IBM's operating system, although it was a very discrete recognized component. You might compare it to the browser that Netscape built and Microsoft's browser which is part of Windows. It was a very discrete piece of software. TSO was a very discrete piece of software that IBM attached to their MVS operating system, and they were giving that away for free and that was part of the basis for ADR's anti-trust suit in 1969. In any event ADR settled this suit in 1970, and Roscoe was very directly competing against TSO, which was free. So ADR was one of the few companies that was going head-to-head against IBM. As I said there was a competitor against the CICS teleprocessing monitor but IBM had priced CICS separately. So I would say ADR was probably the only one competing against free software.

I think everyone recognized that IBM was a potential threat. ADAPSO did have a history with IBM's service bureau, which IBM was forced through a consent decree in 1956 to make a separate company which was called The Service Bureau Corporation. So ADAPSO had a history of being, I wouldn't call it anti-IBM, but the service bureaus were able to compete against IBM because IBM was forced through a consent decree to have a separate service bureau, with a separate name, separate identification. So ADAPSO had a history against IBM's strong monopoly position in different areas and became a good vehicle for ADR to fight its battles with IBM.

## **SUING IBM**

**Yost:** Was there any apprehension on your part or others at ADR in filing suit against IBM given their very extensive legal resources? Did you feel that with the 1956 consent decree and the Justice Department's efforts, you were on very solid ground?

**Goetz:** Yes, we felt on very solid ground. We didn't know if we could get any type of injunction and we didn't quite recognize the costs that were involved in an anti-trust suit. We did have the government in there also suing IBM. There were leasing companies that sued IBM. There were peripheral companies that were suing IBM. So we went in on the basis that we felt that we were being hurt and didn't quite know where it would take us. As it turned

out we settled within sixteen months of filing our suit so it wasn't a long period - from April 1969 when we filed our suit to August of 1970 when we settled out of court. It was a sixteen month period, but had we not settled I don't know that we could have carried the suit too much longer.

## **MANAGING ADR'S SOFTWARE PRODUCTS DIVISION**

**Yost:** When did you take over the Software Products Division, and can you briefly describe your management style?

**Goetz:** I was always responsible for the development and the support of the products, but it was when Dick Jones left in early 1970 and John Bennett became president that I really took it over. John moved the marketing and sales responsibility to me and we made a division out of the software products group and decided not to continue to fund some other projects that Dick Jones had funded in other offices. It was the first time that I really had control of the sales, marketing, development and support. So it was at the beginning of 1970 that I felt I had control of all the resources within the group.

As far as my management style, from 1960 through about 1970 we lost virtually no people. I knew all the people and I built up a staff of maybe thirty to forty people. By 1970 I knew them all very well having worked with them, probably having interviewed just about every one. I would characterize my management style, as Teddy Roosevelt's style, of talk softly and carry a big stick. I never threatened anyone, never got into tirades, thought of the people as my equals. It was very easy to manage the development people. I knew where we were trying to go. I tried to manage through consensus for the most part. I'd say managing the sales people and the sales managers was probably the hardest. I got a lot of help from John Bennett. They were the ones that were always unhappy that products were late and didn't have every feature they would like to see. But I learned how to work with the sales people and always had good sales managers, so I didn't have to direct the sales people per se. We built a loyal staff of technical and sales people. I was very pleased. Over time we lost very few people until we got much larger. At one point we could say that we had never lost a technical person; we did over time, but very few.

## **ADR'S STOCK PRICE**

**Yost:** In the first half of the 1970s profits continued at ADR and new products were being developed and marketed, but by 1974 the stock plummeted from a high, roughly a half decade earlier of forty dollars a share down to a low of one dollar in that year. What accounted for this?

**Goetz:** Well, there was a recession, number one. IBM's unbundling didn't produce lots of profits for software companies because users weren't yet inclined to buy software. They weren't acclimated to it the way they are today. So there were a variety of factors. Number one I'd say the recession; number two was perhaps disillusionment on the potential of software product companies. So our stock, just as it went unrealistically high, it went unrealistically low. I always had confidence that we'd do well, and we weren't doing badly. We didn't have any additional stock offerings at that time and our Software Products Division was profitable, although the company was not. I'd say that 1970, at the time we

settled with IBM, was probably our low in terms of cash, so the IBM settlement was very helpful. And then I'd say it was primarily the recession and perhaps the fact that some of ADR's products were not selling as well as we had hoped; but many software companies, as well as other public companies stock were very low. I don't know that I would say that the Internet company stocks are at their bottom. But we weren't quite in the craze that there is today or there was several years ago with the Internet stocks, but the ADR stock was low because it had a valuation of just a couple million dollars and the company probably had somewhere around \$10 million dollars per year of revenues. It had a valuation of significantly less than ten million.

**Yost:** Did you feel at the time that analysts weren't giving proper respect or attention to the company?

**Goetz:** Well the analysts follow the areas in the industry that are hot and I'd say they stopped following the software industry for a couple of years. There were some accounting questions. We were able to capitalize a lot of software developments. There were some questions about whether the accounting for software was good, similar to what is happening today to Enron. So there were a lot of factors, but the analysts had lost interest in the early 1970s. It wasn't until the mid to late 1970s that the analysts were beginning to follow ADR's stock.

**Yost:** Was the primary accounting issue the amortization of software development costs over too long a period?

**Goetz:** Well I think one was whether you should capitalize development. Machine tools in other industries were capitalized. It wasn't unusual, but there was a certain mystique to software and the question was whether it was like capital equipment that should be capitalized and then amortized. It always was a question from the late 1960s, when it started being approved by FASB, the accounting board. So there was a question whether it was an acceptable accounting practice. At ADR the products were amortized over a seven-year period which turned out to be probably a short period because lots of software have lives much longer than seven years.

## **SOFTWARE ENGINEERING**

**Yost:** In 1968, at the NATO conference in Garmisch, the term software engineering came into use. How much of the efforts to elevate the relevant status of programming do you see as tied to legal issues?

**Goetz:** I don't think it was legal issues. I think that there were lots of large applications starting to be built in the late 1960s and there was a feeling that there was really no systematic way of building software. There were no standards and there were not even guidelines on how to build a software system. I've always compared building a large application to a building. I mean you could be building a straw hut, you could be building a two-story building, you could be building a skyscraper. Certainly when you think of a skyscraper you think about planning for the building of a large structure. And there were a lot of large applications being built by commercial customers, by the government, and there was no systematic way of building them. There was certainly a discipline for building



hardware - for building machines. So I think it's an excellent term, engineering, and you know over the years software design methodologies became popular, but in the commercial world there is still a question of how you go about building a large application. Every company will do it differently.

But I think it became obvious to me, and I think to many others, that software will have a long life. Certainly much of the commercial software and government software that was being built would have a long life and the question was not just making sure you have the reliability and it's bug free when it's first operational. You have to make changes, maintain it, enhance it, how do you go about doing that systematically. And I would say it's still a problem, in terms of the reliability of software. So those were excellent conferences. There probably should be more of them. But the hardware manufacturers, which basically control the programming languages, were less concerned about how the user built his application. They were more interested in selling the hardware. There was no real force around to worry about how to build reliable applications. And many applications that were built turned out not to be reliable or even feasible, and there was a real disenchantment with computers over the years. And there still is within the top management of large companies. But in any event, I always compared building software to building a machine. When you build a machine you really have to think about a lot of things besides getting the machine to be operational.

**Yost:** In terms of relating it to engineering and a machine wasn't that in part to combat, at least for some people, the notion of "software as mathematics" which would prevent its patentability?

**Goetz:** Well, those were the people that were interested in intellectual property and how you protect your assets. I'd say the software engineering conferences were organized purely so that you could build better, more reliable, more maintainable applications. In a certain sense applications were harder to build then because you had smaller memory, you didn't have a vast amount of disk storage. You didn't have networks, you didn't have as many different environments or incompatible hardware that you might have to hook up with. So some of the problems, in a certain sense, are more difficult today. How to build large, what they call today mission critical systems, how to build them reliably was a big question mark then. And those I think were the reasons for the engineering conferences of which there were two in the late 1960s.

## **SOFTWARE PATENTS REVISITED**

**Yost:** You wrote in your memoirs that all the cases coming before the CCPA and the Supreme Court were not representative of the types of program processes software companies were trying to patent. Why was this the case?

**Goetz:** The cases were not representative because they were industrial companies that had filed for patents and as part of the patent there was a computer. But they were not software companies that were filing for patents. These patents were usually controlling the machine, using an industrialized process. And a part of that industrialized process for instance, like curing rubber, included a computer program. So as part of the patent there was a computer involved. However, very often very weak patents get accepted by the patent office. Some that get rejected are never appealed and some just come out of the woodwork. Suddenly

there is a patent before the CCPA or the Supreme Court that has nothing to do with the software business, except one of their claims has to do with an issue that involves software. And in each case, the cases that came before the Supreme Court, they were asking a broad question. Not just should this patent be issued, but should software be patentable. And those were very broad questions that came before the Supreme Court. Often the Supreme Court won't hear a case unless it's broad and when they hear it, they of course make a decision. Unfortunately what happened is, even though the questions were broad, the decisions were always very narrow. So a particular Supreme Court decision did not forever decide the issue that software companies were interested in. But in any event from 1968 to 1980, for a twelve-year period, the Supreme Court and the CCPA were looking at broad questions and never made a decision that was clear to the patent office, whether software had patentable subject matter.

### **ADAPSO AND IBM**

**Yost:** Can you evaluate how successful ADAPSO was at addressing anti-competitive practices at IBM?

**Goetz:** Well, I thought they were reasonably successful. As it turned out, up through 1982 when the IBM Justice Department case was dismissed, IBM for the most part was on their good behavior. They were in court against the Justice Department, there was ADAPSO; they were basically a hardware company. I think they had a certain way of doing business. They had account control through their hardware sales. It was very difficult if you were competing against IBM. But most companies were not, and for the most part IBM was on their good behavior. I think IBM for the most part was more concerned with their hardware competitors than their software competitors. So I think ADAPSO did a very good job. I think, one, software companies were trying to establish themselves as places to buy software. The large manufacturers were very often trying to sell the solution – buy my hardware, buy my software. But I would say that IBM was not doing too many overt things that were really hurting the software industry. ADAPSO, where they saw violations, were quick to point them out to IBM, and I think that was effective in getting IBM to try to be responsive. IBM was also very very large and very often couldn't control their sales force. But I would say for the most part ADAPSO was effective.

**Yost:** More specifically, can you speak on the impact the ADAPSO position papers had on IBM's behavior – did they respond when the position papers were issued?

**Goetz:** Well, they didn't respond publicly, but I would say privately they would try to meet with ADAPSO management. I attended several meetings with IBM, which they had at IBM headquarters, more in the 1980s when we were involved with source code issues. But within ADAPSO we set up an IBM vendor relations committee where IBM personnel would meet with ADAPSO personnel to try to resolve problems, the availability of information, if we could be interfacing to the IBM software. We wanted to make sure we could get that information early. I'd say IBM was careful not to do pre-announcements, but they felt they had a need to tell customers what was coming. I don't think that they were doing things to freeze the market. But we did make them very sensitive and I would say, for the most part, IBM tried to be responsive. But it wasn't a universal problem, in that most software companies were not competing with IBM, because IBM was not in the applications software

business. So all the application software product vendors really had virtually no problem with IBM and it was only the system software companies, some of us had problems with IBM. I'd say the biggest problem was later on in the 1980s after the IBM Justice Department suit was dismissed and when IBM was losing market share with IMS, that they got very aggressive with their DB2 product and the DBMS software vendors really were concerned about IBM's business practices.

### **OTHER ADAPSO ACTIVITIES**

**Yost:** This was one area of ADAPSO you were very much involved in. Were there other committees or work within ADAPSO where you also became involved?

**Goetz:** I was involved with the tax group, in which I probably was in a minority, because I took the position that software should be taxed like any other product in the marketplace. There were taxation issues, there were accounting issues, revenue recognition issues. Part of ADAPSO's goal was to educate new members and smaller companies so there were some president's councils, where company presidents could meet in roundtables where they could discuss common problems. So ADAPSO was much more than a group fighting IBM. There were lots of issues that had nothing to do with IBM, but were association issues which benefited companies that belonged to ADAPSO.

### **PRODUCT DEVELOPMENT AT ADR**

**Yost:** Can you describe how product development practices at ADR changed from the early years, say from the late 1960s to the late 1970s?

**Goetz:** I'd say starting in the mid to late 1970s we always had two-year development plans. Each group would have to provide once a year a two-year development plan, and usually try to plan for two releases - in other words the upcoming release and the release after that. We put out at least one major release between once a year and once every eighteen months. And we thought of these products as having long lives, which many of them did. So we had a very formal planning process. We had user group meetings where we would tell our users what was coming in the next version. We had feedback from our users to make sure that we were building the right facilities.

In the late 1970s we started integrating our products which we thought was a good strategy, quite different from bundling because you can buy these products separately; bundling is not illegal unless you're a monopoly and you're tying it. But as the company grew we felt, from a marketing perspective, that our users wanted to buy many of our products but they didn't want to worry about how to get them to talk to each other. So there was lots of planning to be done as the company got bigger. It got a little too big for me and I made sure I had good staff to coordinate the integration of our products, which meant working with the major programming groups within ADR to make sure the products were properly integrated.

### **ACQUIRING DATACOM**

**Yost:** Getting into the database management area with the Datacom acquisition forced ADR to change its sales and marketing practices. From a revenue standpoint it was a very successful acquisition. Were there any regrets to broadening into this area?

**Goetz:** I sort of regretted it as I looked back. But it was an exciting period. One, there was a lot of new developments in database management systems. We called ourselves a relational database, which was a new type of database. We integrated the database products with some of our other products. Database management was an area that users were very interested in. We grew when we acquired the Datacom products from the Insyte Corporation. It had about \$2 million in sales in about 1978 and by 1985 or 1986 it was probably about seventy or eighty million dollars. We grew that business quite fast. The whole company grew with it. And although our profits did go up, it did really change the nature of the company because it was much different from selling programming tools, where you would sell to middle management. Here you would be selling to top management. It was a very long selling cycle. It was a sale anywhere from several hundred thousand dollars to up to a million dollars, and it changed the whole nature of the company. It certainly put us on the map. We grew much faster but it took some of the fun out of being a small company. We ended up being quite a large company and ending up being very much in direct competition with IBM. Over time we had learned to live with competing against IBM's free TSO and ADR was doing very well with Roscoe. In certain respects I was unhappy that we were in the database management business, but once being in there it was very exciting and we had a lot of growth, but we were also head to head against IBM, which was something I really wanted to avoid. We ended up for several reasons being in that business. Part of it was John Bennett's belief that as a public company we needed to grow faster and so we ended up being in the DBMS business, with a good product, but it changed the nature of the company significantly.

## **THE IBM AND MICROSOFT SUITS**

**Yost:** In your opinion, how similar are the fundamental issues in the government's case against the practices of Microsoft in the 1990s compared to IBM and the Department of Justice's case against IBM in the late 1960s and 1970s?

**Goetz:** I think there are lots of similarities, but a lot of what IBM did was to unbundle their applications from their operating system software in the 1970s and unbundle their operating system software in the late 1970s, which they did voluntarily. So you don't have a case of tie-in sales per se with IBM, which you do have with Microsoft. IBM never had this great desire to put applications into the operating system. IBM was still fundamentally a hardware company and I don't think it was out to necessarily capture 100 percent of the software business. Microsoft, I'd say, would like to capture as much of the software business as they can and I'd say is much more unscrupulous in terms of their view of what they consider to be legal. Their position, and Bill Gates' position, is that anything under the sun should be in the operating system. Certainly tie-in sales are considered illegal per se under the anti-trust laws. So I don't think Microsoft's positions are valid. I think they are out to destroy software companies so they can capture just about 100 percent of the market if they can. I look at both organizations quite differently. Some of the issues were the same. Certainly the availability of interface information, pre-announcements, tie-ins, are all similar issues but in terms of the companies themselves I'd say they're quite different.

**Yost:** Had Microsoft been forced into splitting into two independent companies, one for applications and one for operating systems, would that have been an effective remedy and good for the software industry in your opinion?

**Goetz:** I definitely think it would have been good for the software industry. Unfortunately that is not going to happen. I think at this point Microsoft is speaking a lot of double-talk when they say that if they are going to be restricted in any way, that they are not going to be able to innovate. I think, if there were two separate companies, it would benefit the software industry. It would certainly clarify things if they actually competed against each other. Of course the question would be, it's all theoretical, if they control the operating system company and said that a particular application should be part of the operating system such as a media player, which is in question, or other pieces of software, who is to decide whether it is an application or whether it is part of an operating system. To the extent the Microsoft application company would need interface information and other things, they'd just be another company in theory asking for information from the Microsoft operating system company. So it certainly wouldn't hurt. Whether it would solve all the problems that are facing software companies today is unclear. But it certainly wouldn't hurt software companies or hurt Microsoft which would still be successful in my opinion.

## **PROGRAM DEVELOPMENT TOOLS**

**Yost:** In your memoirs you said that languages and programming tools haven't advanced in a way to accomplish all they perhaps could. Would you comment a little further on that?

**Goetz:** Well, I've been in the computer field now almost fifty years and I saw what it took to program an application back in the 1950s. And I also saw the hardware advances. If you look at where hardware has come from, a Univac I or II in terms of capacity being a lot less than a \$600 PC, you can see all the advances in hardware. And yet when you look at programming and programmer productivity and how a programmer goes about developing a large application, one has to be very frustrated. In the 1970s I saw the evolution of assembly language to COBOL and then in the 1980s I saw the evolution of a lot of applications being programmed in what they would call 4GL's, forth generation languages. And ADR had built IDEAL, which was a 4GL, and I kept thinking that we'd see the evolution of programming languages that would keep pace with the advances in hardware. But when I see programming being done in JAVA and C++, and although there are components and other tools available to help in programming, I don't see the advances in programmer productivity that I think should have been possible. So there is just a frustration. Object-oriented programming, which I was very critical of in the late 1980s and 1990s, I don't think OOP has necessarily made it easier to program or made programs more maintainable. So I don't have the answer, but I would say there is still a lot of programming tools that are needed and I would hope over time that we could look back and we see the evolution to higher level programming languages so that programs can be built faster and be more reliable and more maintainable than they are today. Certainly if you talk to large corporations, there is still the frustration that building computerized mission critical applications are very very difficult and that's why many of them outsource today. But the outsourcing companies are also having difficulties building large applications. Granted, problems are more complex today. We have networks,

we have the Internet, we have more complex applications we are trying to build, but still I would think there's a need for better programming tools and better programming languages.

**Yost:** Do you see any policies that could be implemented that might create an environment more conducive to such innovation?

**Goetz:** Well, one policy is to make sure there is a healthy independent software vendor community. I think you would see a lot of innovations coming from ISV's. I think making sure there are standards committees to the extent that the government, the National Bureau of Standards years ago got involved with computer programming languages. The government, if nothing else, which is a very, very large user of computers, should make sure that there is research being done into better ways of programming. Certainly at one point they thought ADA was going to be the godsend, which apparently was good for government applications but not necessarily as a programming language for commercial applications. But I do think there is a need for better programming languages, better programming tools, and for that you need a competitive environment. For that reason I think the government should make sure there is a healthy software industry.

**Yost:** Okay, well thank you very much for your time this morning.

**Goetz:** My pleasure.