Memoirs of a Software Pioneer: Part 2

Martin Goetz

In the second of this two-part series, these memoirs feature Goetz's role in growing ADR to a $200,000,000 company and in ADAPSO, which dealt with industry issues and fought for a level playing field as IBM began to exert its enormous marketing powers and resources in the licensing of its program products.

Part 1, published in the Annals' *Jan.–Mar. 2002 issue, traced the software industry's beginnings, particularly the author's experience as a founder of Applied Data Research in 1959 and his involvement with intellectual property and antitrust issues. These issues led to IBM's unbundling of software and hardware in 1970.*

In August 1970, Applied Data Research (ADR) settled its antitrust suit against IBM out of court and reached a temporary peace with IBM. Although the ADR/IBM suit was over, there was no agreement that would stop ADR or me from working with the US Department of Justice (DOJ) or a trade organization to try to curtail IBM's possible dominance of the emerging software industry.

Since 1968, ADR had been active in the Association of Independent Software Companies (AISC), a small software trade association. At the time of the IBM/ADR settlement, I was serving as ADR's representative. At about the same time, the Association of Data Processing Organizations (ADAPSO), a trade association of small service bureau centers, was looking to expand its scope and so opened discussions with AISC.

ADAPSO and the software industry: 1970–1975

ADAPSO asked Larry Welke, International Computer Programs (ICP) president, to establish its software entity. In October 1971, ADAPSO's Software Section was formed with 26 software companies and Welke as its first president.

I had known Larry for several years, and we quickly pursued discussions about AISC's joining ADAPSO. This became reality in the spring of 1972, and we named the expanded section ADAPSO/AISC. It was a harmonious merger. AISC and ADAPSO had already worked closely together during 1971 and 1972. We had visited the DOJ together, and both organizations were following the IBM/DOJ antitrust suit carefully. ADAPSO's membership—with its original makeup of independent computer service companies and its newly formed Software Section—now had a compelling interest in both hardware and software.

The ADAPSO/AISC merger, coupled with ADAPSO's early efforts against IBM, signaled the software products industry's true beginning. The question was if IBM would dominate this new industry, smothering ADR and the hundreds of other newly formed, independent software companies. Of course, we didn't know the outcome then, but through the power of trade associations, we helped equalize the playing field.

In February 1972, ADAPSO held a two-day economic meeting and addressed IBM's dominance in the hardware industry. IBM, an ADAPSO member, drew repeated criticism. Bernard Goldstein, then ADAPSO president, spoke about the ways in which IBM used unfair selling and service practices to compete with its computer services customers, and he expressed his consternation with IBM for failing to participate in the meeting. He also criticized the government for not recognizing the service companies' enormous potential and the computer industry's changing nature.

ADAPSO kept the heat on, filing a July 1972 petition in federal court asking to be heard in the IBM/DOJ suit, charging civil rights violations and asking that the government lift its ban of secrecy surrounding its case against IBM. Although ADAPSO was told it had "no standing," presiding Judge Edelstein immediately lifted the secrecy ban and gave credit to ADAPSO for protesting the ban. I attended the meeting, where I met Milt Wessel, ADAPSO's counsel and former federal court prosecutor who had prepared the argument. Over the next 19 years, Milt and I worked closely together, until his death in 1991.

As part of the merger, I was named to the board of directors of ADAPSO's new Software Section and asked to form a committee on software protection. In late 1972, I was elected as the 1973 president of ADAPSO's AISC sec-

1058-6180/02/$17.00 © 2002 IEEE

tion, now renamed ADAPSO/SIA (SIA stands for Software Industry Association).

ADR was also a member of the Communications and Computer Industry Association (CCIA), a group formed in 1972 specifically to fight IBM's monopoly of the computer industry. John Bennett, ADR's president, became ADR's representative at CCIA, and I represented ADR at ADAPSO.

ADR remained aggressive during this period. Having settled the IBM suit in August 1970, ADR had no time to lose. We knew we could not stop IBM from bundling its Time Sharing Option (TSO) with its Multiple Virtual System (MVS) operating system—because that was the essence of ADR's out-of-court settlement—but there was no reason that IBM should continue to bundle its operating system with its hardware.

We were also concerned about IBM fairly pricing its new software offerings. In 1972, ADR wrote to the DOJ complaining that IBM's bundling of its operating system software was illegal, inhibiting the growth of ADR and the computer software market. We were sent a sympathetic letter in which the DOJ sent us a copy of its "Preliminary Memorandum on Relief," should the department win its case against IBM. The relief covered many issues that would benefit the software industry, but who could wait? The only other alternative was seemingly in the court of public opinion.

In October 1972, *Datamation* published an article I wrote, "The Monopolized Software Industry." In it, I appealed to the DOJ for quick action and I also attacked the computer manufacturers about steadily rising costs for end-user development of computer applications. Since my early days in the computer field, I had fervently believed that higher-level programming languages and programming tools were the key to reducing both the programming backlog and programming's cost. This belief was reflected in just about everything I did and in the software that ADR was building. It was also reflected in the articles I wrote and, indirectly, in my battles with IBM.

The DOJ's 1969 complaint stated that "IBM had inhibited the growth of software companies." It was therefore appropriate for ADAPSO's new SIA software section to develop a position paper recommending changes in IBM's policies and practices. IBM's voluntarily unbundling of its software in 1970 was a step, but much more was needed. So, shortly after the AISC/ADAPSO merger in 1972, I joined a new ADAPSO committee to draft a position paper on the relief software companies required to ensure a viable software industry.

Our committee, called the Economic Study Group, met twice in late 1972. (A guest at one meeting was Al McCallister from the DOJ's Antitrust Division.) We developed a working draft—"IBM's Monopolization of the Software and Services Industry"—that in February 1973 was ready for approval by the ADAPSO/SIA and ADAPSO Boards of Directors.

As president of the ADAPSO/SIA section and chairman of ADAPSO's Software Protection Committee, I championed approval of the position paper. Its recommendations were specific.

- It covered the ways in which IBM's policies, including its strong account control,[1] were hurting IBM hardware users.
- It recommended that IBM be split into a hardware company and into a software company similar to IBM's Service Bureau.[2] The software company would have an entirely separate name, facilities, personnel, and board of directors.
- It proposed specific relief covering early release of interface information, pricing to reflect full administrative and marketing costs, and product announcements (no product pre-announcements allowed).
- It proposed the unbundling of all software, including operating systems.

All of ADAPSO's boards endorsed the position paper, despite the controversial proposal for a separate software company. The paper was distributed to the press and widely covered.

ADR recognized that this position paper would not, in itself, change IBM's practices but believed it could influence the DOJ and large IBM mainframe users, ADR's primary prospects. Publication of this paper also made the world more aware of the Independent Software Vendors (ISVs)—those thousands of software companies that had emerged as soon as IBM unbundled its software in 1970.

The events of 1973 signaled a turning point in my career. Although I had worked in the computer field for more than 18 years, it was largely in supervising technical jobs, writing articles, and dealing with industry issues. I was now taking a leadership role in the industry and starting to feel comfortable operating ADR's Software Products Division where I had profit and loss responsibility and complete operational control over the product development effort, support, sales, and marketing. The year 1973 was also a personal turning point. In nine short months, I had married and gained a family with my wife's two young daughters.

Although a public company, ADR was still small and struggling to survive. The US was in

the midst of a recession, and large-company IT departments were slow to purchase software products from ISVs. The presidency—which had a one-year term—of the ADAPSO/SIA section gave ADR and me a great deal of public exposure. The ball was now in my court to help ADR grow.

ADR'S growth and products: 1970–1975

Besides fighting IBM and going to trade association meetings, John Bennett and I were both actively working to ensure ADR's survival. In 1970, ADR's Software Products Division (SPD) accounted for about 50 percent of ADR's revenues. I made it a flat organization structure with four product managers, two sales managers, and several administrative groups reporting to me. Because we had no formal marketing department, I coordinated all marketing and PR activities.

After settling the IBM suit, Bennett and I focused on the products developed in Princeton, New Jersey, and halted development in our other offices. Some of the Princeton products were selling well, but others weren't. I had great confidence in all the products we had developed in Princeton in the 1960s, and Bennett supported my position that, in time, all would be successful.

ADR had a strong, stable development staff. Bob Wickenden and Steve Wright—both ADR founders—remained strong programmers for their entire careers. During the 1960s we retained almost all of our professional staff, a group I had personally interviewed and knew well.

I spent much of my time trying to influence the direction and evolution of ADR's program development tools. I had developed a strong interest in tools dating back to my programming experiences in the 1950s, and I believed the market for such tools was largely untapped. I had also seen computer languages evolve for developing commercial applications. In the 1960s, hardware companies were promoting Cobol as a third-generation computer language and as a replacement for second-generation assembly programs. IBM and other hardware companies provided a free Cobol compiler with their hardware but offered little else related to program development tools.

In the late 1960s, ADR's Princeton program development products were Autoflow, the Librarian, Remote OS Conversational Operating Environment (Roscoe), System Analysis Machine (SAM), and MetaCobol, a Cobol preprocessor. Autoflow, ADR's first product, had emerged primarily because in 1963 RCA was interested in such a program for its users. ADR built the Librarian in 1965, on the other hand, to maintain the Autoflow source code on tape and then commercialized it in 1967.

In 1970, ADR was financially precarious but continued to enhance its products. We built major enhancements to Autoflow, priced as extra cost options, and in 1973 announced Autoflow II with Automated System Charter (ASC) for system charts, Cross Program Analyzer (CPA) for cross-referencing an entire Cobol application, and Extended Text Compositor (ETC) for creating and editing large documents. ADR had built ETC for internal use in developing product manuals; the other options were built from scratch. We also enhanced the Librarian with disk file support.

Expansion for Roscoe and MetaCobol in the early 1970s, on the other hand, was hard to justify. Neither product was selling well, but we were astute enough to realize that they were products of the future.

Roscoe's and MetaCobol's development began in 1967–1968 when Equitable Life, an early Autoflow user, purchased the RCA Octopus computer, a special-purpose hardware-software system using RCA's Spectra 70 Series computer with many online terminals. Equitable Life dedicated Octopus to online Cobol application development for programmers' sole use. Equitable was a large user of IBM System/360 computers.

We quickly recognized that if Equitable could spend hundreds of thousands of dollars for a special-purpose computer for its programmers, a software-only product like the Octopus for IBM's System/360 line of computers might represent a good business opportunity. Such a set of programs would make it significantly faster, easier, and cheaper to develop Cobol applications.

I assigned a new hire, Marty Kramer, to do some research on how we might build such a system. He discovered the Wayne Remote Access Processor (WRAP) program at Wayne State University in Detroit, Michigan, which supported online terminals. The WRAP system could be used for remote job entry, had its own language for writing terminal script programs to prompt users, and performed other functions such as editing and validating. The WRAP program executed under IBM's OS operating system, and we could add Cobol debugging features to interact with IBM's 2260 terminals. ADR bought the rights for the WRAP system from Wayne, and we named the new product-to-be "Conversational Cobol."

Steve Wright, who had worked on the RCA Cobol compiler, was also assigned to the project. He decided that the best way to have the

online terminal system interact with the Cobol program was to add Cobol-like debugging statements to a standard Cobol application for online testing. So he built a flexible Cobol preprocessor in which the user inserted non-Cobol statements and the preprocessor expanded the non-Cobol statements into standard Cobol statements acceptable to the Cobol compiler. This concept resembled macro statements in assembly programs for expanding the assembly language. It soon became obvious that besides expanding debugging statements, the preprocessor could also expand the Cobol language with higher-level Cobol-like statements.

The WRAP system could also be used to debug Fortran, PL1, and assembly programs. Within the year, we decided to separate Conversational Cobol into two products. The WRAP-enhanced program became Roscoe, and the Cobol preprocessor became MetaCobol.

Although both programs elicited great interest, they sold poorly in the early 1970s for several reasons. Roscoe had to compete with IBM's free TSO for program development, and most companies lacked 2260 terminals for their programmers. MetaCobol was too new a concept for most IT professionals, and ADR did not have a large library of Cobol macros. But rather than abort these products, we continued to enhance them. Our gamble began to pay off by the mid-1970s.

Autoflow continued to sell well, but the Librarian became the big seller in the early 1970s as companies recognized that it was irresponsible to maintain source programs on IBM cards. Additionally, Pansophic entered the source program maintenance and security market, and ADR and Pansophic dominated and shared that market for many years. It was a classic example of competition stimulating the marketplace. It was not a question of should I buy such a program, but which program should I buy?

As ADR's SPD president and director, my management style was the classic "manage by walking around," which was why I had always situated my office near programmers and technical support personnel. I frequented the coffee room many times each day, conversing with all levels of personnel on my way for yet another cup. I also typically dropped into a product manager's office unannounced.

The four product managers (Bob Caughey, Dick Kauffman, Phil Berg, and Bob Jones) were responsible for all development, support, packaging, and documentation of their products, and for acting as liaison with the sales organization. I worked closely with all of them for many years.

When Bennett became president of ADR in 1970, responsibility for the domestic and international sales organization fell to me. With Bennett's help, I learned how to direct them. Salespeople, a breed apart from programmers, always questioned why products weren't developed faster, or why a product didn't have every bell and whistle. But we had strong sales plans to keep them motivated, and the products had strong user acceptance.

Bennett and I shared ADR's management. He ran all the other divisions of ADR, which in the early 1970s consisted of a time-sharing division, two programming services divisions, and a control systems division. He also had legal, financial, and facilities responsibilities that were part of the corporate office. Because ADR was a public company, he was also our point man for Wall Street contacts. Depending on the year, there was either some Wall Street interest in ADR or none at all. ADR's stock, which had peaked at $40 in 1968, had sunk to $1 by 1974.

Ongoing software protection issues: 1970–1975

My involvement in software protection continued. Now I was actively writing articles for the trade press on the nature of software and the need for software protection—a concept still in a state of flux and confusion. The terms *software protection* and *intellectual property* (IP) today are used interchangeably. But in the early years, the definition of software was unclear, as was the question of software's protection by IP laws.

With IBM's unbundling, the IP issues had multiplied, become more complex, and all were controversial. The question, Is software patentable? which started about 1965, continued well into the 1970s with narrow and confusing decisions by the Supreme Court. The Patent Office was no help. Each new commissioner would issue different software patenting guidelines and different positions on whether software processes were patentable. The controversy on whether patents, copyrights, or trade secrets were the best form of protection also continued. Additionally, now that software was a commercial, licensed product, there was much controversy on whether software was taxable. The determination of software as tangible or intangible affected its tax status. And, more importantly, the question of existing state and federal law applicability remained unanswered.

So during 1970–1975, I spent a great deal of energy on software protection issues that were beginning to receive national attention. My involvement with one committee in the early 1970s was particularly memorable. In May 1970 (during ADR's antitrust trial against IBM), ADR was invited by the National Council of

Patent Law Associations (NCPLA) to join a new committee it was forming, the Committee to Study Computer Program Protection. Its purpose was to explore the need for data processing software to have additional statutory protection.

I represented ADR on this committee. It was a prestigious group, consisting of 32 members from industry, law firms, academia, and the government, and chaired by Harry Mayers, former chief patent counsel at General Electric (GE). Members included senior people from IBM, Honeywell, Burroughs, the Patent Office, the Copyright Office, the DOJ, the National Science Foundation, the National Bureau of Standards, and the American Federation of Information Processing Societies (AFIPS), along with eight law firms, three universities, two trade associations, and eight representatives (including ADR) from the software industry.

Originally, the life of the committee was to be one year. At the first meeting in August 1970, a subcommittee was formed to gather data on the software industry in general and on the present state of protection for computer programs. This work was to be a prelude to drafting new legislation. Over the next two years, the group never reached consensus on whether new legislation was needed and what form it should take. The chairman recognized the difficulty of reaching consensus with such a diverse group, as well as the problem's complexity, and several times tried to dissolve the group or bring in a new chairman.

In June 1972, with little to show in terms of a consensus on the necessity for new legislation or on the relevancy of existing laws, the group considered disbanding. Some of us argued that existing patent law was adequate, while IBM and others proposed a program registration system to replace existing patent law. Regarding copyright law, there was more of a consensus that perhaps new (or clarifying) legislation was needed. In September 1972, I wrote an open letter to the committee urging it not to disband and expressing the need for such a committee to protect software companies and their software assets.

By November 1972, the group was disbanded, having failed to come up with a consensus on either the need for new legislation or its form. Today, as I reflect on these early exercises of mine to help establish software protection for ADR and the software industry, I am surprised at my lack of reticence. Being an active NCPLA committee member was a good education for me, and many of its members later became valuable contacts.

In the early 1970s, I had a clear vision and a passion for what I believed was the nature of software. It evolved from the 1960s when ADR and AISC filed an amicus brief in the famous 1968 Prater-Wei patent case argued in the Court of Customs and Patent Appeals (CCPA). We argued in that brief "on the equivalence of hardware and software and that a machine process patentable as hardware should be equally patentable if implemented in software (a program)."[3] Over the years, that argument became my mantra. I had also read the proceedings of the NATO Software Engineering Conference held in Europe in October 1968. Back then there was a growing awareness of the difficulty of designing and building large, complex software systems. My reading reinforced my belief that software development was a science and that we were building software machines. The terms *machines*, *engineering*, and *software* became linked, affecting my thinking and approach to software development and to software protection.[4]

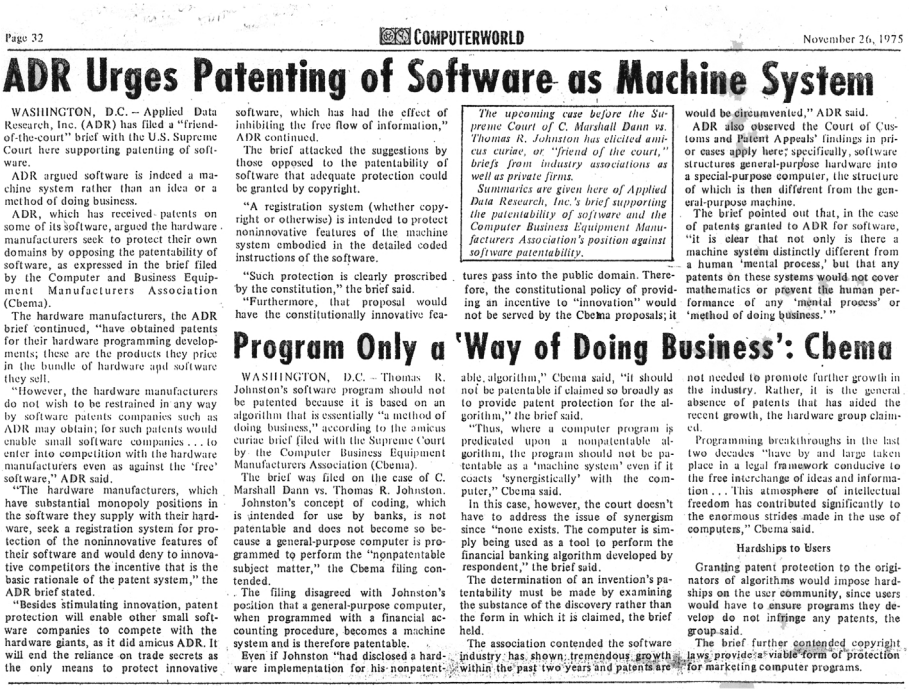
But no one really knew if the existing copyright and patent laws covered computer programs, and the industry expected that the Supreme Court would have the final say. Unfortunately, the cases before the CCPA and Supreme Court were not representative of the types of program processes (within computer programs) that software companies were trying to patent.

When I joined ADAPSO in 1972, I chaired the Software Protection Committee and asked ADAPSO to file an amicus brief in the Benson-Tabbott Supreme Court patent case that AISC had agreed to support. ADR was also filing an amicus brief for this case, and ADR's patent attorney, Mort Jacobs, wrote both the ADAPSO brief and the ADR brief. In this 1972 case, Bell Labs was trying to patent the binary-coded-decimal-to-pure-binary-processing logic, using a conversion program. This patent probably should never have been filed or appealed. The courts had said that mathematics and laws of nature were not patentable, and the Benson-Tabbott patent was close to a mathematical algorithm. In the ADAPSO and ADR amicus briefs, our attorney had used handwriting recognition and voice recognition as the types of programs that might contain patentable processes. Certainly, the unique process of analyzing handwriting is worthy of patent protection, we argued. Ironically, although there are now many crude handwriting recognition patents, that problem is still looking for a better solution.

In 1972, there was hope that the Benson-Tabbott patent case could resolve the issue of software patentability. The narrow Supreme Court decision, unfortunately, meant that the

controversy continued. It was one of many cases filed before the Supreme Court, and ADR, ADAPSO, and I remained involved in the software patentability issue.

In 1975, ADAPSO and ADR each filed another amicus brief before the Supreme Court in the *Dann* vs. *Johnston* patent case.[5] I attended the oral arguments and hoped that the Supreme Court would recognize that it was expected to rule one way or another on software patentability. But it was not to be. Again, the Court made a narrow decision rejecting the patent on "obviousness" and did not address whether software processes were patentable under existing patent law. (Not until 1981 was the issue resolved in favor of patenting software processes.)



Page 32 COMPUTERWORLD November 26, 1975

ADR Urges Patenting of Software as Machine System

WASHINGTON, D.C. – Applied Data Research, Inc. (ADR) has filed a "friend-of-the-court" brief with the U.S. Supreme Court here supporting patenting of software.

ADR argued software is indeed a machine system rather than an idea or a method of doing business.

ADR, which has received patents on some of its software, argued the hardware manufacturers seek to protect their own domains by opposing the patentability of software, as expressed in the brief filed by the Computer and Business Equipment Manufacturers Association (Cbema).

The hardware manufacturers, the ADR brief continued, "have obtained patents for their hardware programming developments; these are the products they price in the bundle of hardware and software they sell.

"However, the hardware manufacturers do not wish to be restrained in any way by software patents companies such as ADR may obtain; for such patents would enable small software companies . . . to enter into competition with the hardware manufacturers even as against the 'free' software," ADR said.

"The hardware manufacturers, which have substantial monopoly positions in the software they supply with their hardware, seek a registration system for protection of the noninnovative features of their software and would deny to innovative competitors the incentive that is the basic rationale of the patent system," the ADR brief stated.

"Besides stimulating innovation, patent protection will enable other small software companies to compete with the hardware giants, as it did amicus ADR. It will end the reliance on trade secrets as the only means to protect innovative software, which has had the effect of inhibiting the free flow of information," ADR continued.

The brief attacked the suggestions by those opposed to the patentability of software that adequate protection could be granted by copyright.

"A registration system (whether copyright or otherwise) is intended to protect noninnovative features of the machine system embodied in the detailed coded instructions of the software.

"Such protection is clearly proscribed by the constitution," the brief said.

"Furthermore, that proposal would have the constitutionally innovative features pass into the public domain. Therefore, the constitutional policy of providing an incentive to "innovation" would not be served by the Cbema proposals; it would be circumvented," ADR said.

The upcoming case before the Supreme Court of C. Marshall Dann vs. Thomas R. Johnston has elicited amicus curiae, or "friend of the court," briefs from industry associations as well as private firms.

Summaries are given here of Applied Data Research, Inc.'s brief supporting the patentability of software and the Computer Business Equipment Manufacturers Association's position against software patentability.

ADR also observed the Court of Customs and Patent Appeals' findings in prior cases apply here; specifically, software structures general-purpose hardware into a special-purpose computer, the structure of which is then different from the general-purpose machine.

The brief pointed out that, in the case of patents granted to ADR for software, "it is clear that not only is there a machine system distinctly different from a human 'mental process,' but that any patents on these systems would not cover mathematics or prevent the human performance of any 'mental process' or 'method of doing business.'"

Program Only a 'Way of Doing Business': Cbema

WASHINGTON, D.C. – Thomas R. Johnston's software program should not be patented because it is based on an algorithm that is essentially "a method of doing business," according to the amicus curiae brief filed with the Supreme Court by the Computer Business Equipment Manufacturers Association (Cbema).

The brief was filed on the case of C. Marshall Dann vs. Thomas R. Johnston.

Johnston's concept of coding, which is intended for use by banks, is not patentable and does not become so because a general-purpose computer is programmed to perform the "nonpatentable subject matter," the Cbema filing contended.

The filing disagreed with Johnston's position that a general-purpose computer, when programmed with a financial accounting procedure, becomes a machine system and is therefore patentable.

Even if Johnston "had disclosed a hardware implementation for his nonpatentable algorithm," Cbema said, "it should not be patentable if claimed so broadly as to provide patent protection for the algorithm," the brief said.

"Thus, where a computer program is predicated upon a nonpatentable algorithm, the program should not be patentable as a 'machine system' even if it coacts 'synergistically' with the computer," Cbema said.

In this case, however, the court doesn't have to address the issue of synergism since "none exists. The computer is simply being used as a tool to perform the financial banking algorithm developed by respondent," the brief said.

The determination of an invention's patentability must be made by examining the substance of the discovery rather than the form in which it is claimed, the brief held.

The association contended the software industry has shown tremendous growth within the past two years and patents are not needed to promote further growth in the industry. Rather, it is the general absence of patents that has aided the recent growth, the hardware group claimed.

Programming breakthroughs in the last two decades "have by and large taken place in a legal framework conducive to the free interchange of ideas and information . . . This atmosphere of intellectual freedom has contributed significantly to the enormous strides made in the use of computers," Cbema said.

Hardships to Users

Granting patent protection to the originators of algorithms would impose hardships on the user community, since users would have to ensure programs they develop do not infringe any patents, the group said.

The brief further contended copyright laws provide a viable form of protection for marketing computer programs.

Figure 1. Two opposing software patenting arguments from a 26 November 1975 *Computerworld* article. (Reprinted with permission of *Computerworld* magazine.)

As I reviewed my files for these memoirs, I came across a 1975 *Computerworld* article that had two headlines (see Figure 1). The first—"ADR Urges Patenting of Software as Machine System"—was my mantra for more than 10 years, and I had repeated it many times in speeches and articles. It was our argument in the ADR brief in the Johnston case.

The second headline—"Program Only a 'Way of Doing Business': Cbema"—represented Cbema's (Computer Business Equipment Manufacturers Association) amicus brief position. IBM dominated that association, and this headline represented IBM's opposition to software patenting. Its position was that a general-purpose computer is programmed to perform nonpatentable subject matter. So the battles between ADR and IBM continued in the courts as well as in the marketplace.

More software industry issues: 1975–1980

From 1975 to 1980, I continued to get involved with those industry issues critical to ADR's survival. Just about all my thinking concerned survival, rather than growth. There was real fear of IBM. IBM had a tendency to take giant steps, knowingly, and sometimes unknowingly, stepping on small independent software companies. ADAPSO was a good vehicle for software companies to fight IBM along with the DOJ, which in 1975 was preparing to go to trial in New York City. Since 1967, ADR and I had periodically communicated with the DOJ concerning what we believed to be antitrust violations on IBM's part. During the early 1970s, we had exchanged a lot of correspondence and had occasionally met.

In 1975, I and several other software executives were asked to be government witnesses in the IBM/DOJ suit, and in spring 1976, I testified for 11 days over a three-week period, following Larry Welke. We, and the others, were government witnesses for the software part of the Justice's antitrust suit. Much of my testimony was about my experience in the 1960s, prior to IBM's unbundling. I covered ADR's difficulty in marketing its Autoflow products, and I spoke about the difficulty users had in developing applications because of IBM's free (but error-prone) operating systems and because of the limited number of programming tools available. In 1976, IBM was still providing all its operating system software at no charge, and I was trying to provide testimony that its free operating system software, for which IBM had 100 percent market share, was not in the public's best interest. This was an opportunity to attack IBM and its free operating systems of which TSO was a

part. TSO continued to hurt Roscoe sales.

Naturally, IBM challenged my testimony, wanting me to provide proof. Most of my testimony concerned what I had read in trade periodicals, but almost 10 years later it was hard to be specific. I spent the following weekend reviewing 1960s periodicals in the ADR library. Both *Computerworld* and *Datamation* had extensively covered IBM's operating systems problems. Issue after issue featured war stories on how IBM users were finding errors in IBM's software and having operational problems. When my testimony continued that next week, I had numerous references to point to.

In my testimony, I stated that there were many software companies that could develop operating systems and that such competition could both improve the efficiency of operating systems and reduce the resources required, as well as offer new features.

I also testified about IBM's account control and how this hurt a software company's ability to compete against IBM. IBM's counsel, Paul Dodyk, feigned shock. When he requested the names of the companies where IBM had account controls, so that it could initiate an investigation, the court reprimanded him for attempting to intimidate the witness.

Testifying proved an invaluable experience. I had my opportunity to tell the world why IBM's unbundling was a godsend for the user community, why IBM should unbundle its operating systems, and why an independent software industry would benefit the common good. It was a great experience for me and for ADR.

ADR had other confrontations with IBM during the mid-1970s. ADR had developed the Librarian/online, which had many of Roscoe's attributes but which was for users of IBM's Disk Operating System/Virtual Storage Extended (DOS/VSE) operating system. DOS/VSE was geared toward IBM's smaller users, and users had nothing equivalent to TSO, which was part of the MVS operating system. (In 1970, ADR had settled its Roscoe antitrust suit with IBM and agreed to live with TSO being given away free as part of its out-of-court settlement.)

In the early-to-mid-1970s, IBM also started charging for its field-developed programs (FDPs), which it previously had been giving away free. Two of these were the Entry Time Sharing System (ETSS) and the Customer Information Control System (CICS) Source Program Maintenance (SPM), both for the DOS/VSE operating system. Neither was classified as an IBM program product, and IBM provided only limited support for them. IBM priced ETSS at $250 monthly and waived the charge after 24 months. The SPM charge was $124 per month, waived after 12 months. These low prices seriously affected the Librarian/online sales, and we complained to IBM and the DOJ. We also brought it to the attention of ADAPSO and its IBM liaison, Ed Kane.

Prior to IBM's entry into this market, the Librarian/online, introduced in 1976, had been a big seller. In 1977 ADR announced a new version of the Librarian/online, renamed Vollie (for VS Online Librarian Extended).

ADR met regularly with IBM at ADAPSO and in Princeton on this issue and let it know that, to compete, ADR had reduced the prices of Librarian/online by more than 50 percent. We also maintained contact with the DOJ, suggesting that IBM's monthly pricing of its software products, along with its account control, was unfair competition. Although ADR was unable to stop IBM from marketing its FDPs at these low monthly rates, I believe it made IBM more sensitive to our problems.

In 1979, IBM announced the IBM 4300 as a low-priced departmental computer, repriced a new version, called ETSS/II, from $325 to $65 per month, and renamed it Interactive Computer and Control Facility (ICCF). ADR complained bitterly to IBM, the DOJ, and ADAPSO that this was predatory pricing, unfair competition, and that it would drive Vollie out of the market. But there was little ADR could do but complain.

ADR's growth and product evolution: 1975–1980

By 1975, IBM was pushing terminals and suddenly Roscoe began to sell along with MetaCobol and several other ADR products. The recession was over, and users were more comfortable buying from ISVs like ADR.

ADR, expanding at about 35 percent per year, was now becoming well known worldwide. In 1975, we announced the installation of our 5,000th product and were receiving many Datapro awards for product excellence. By the end of 1977, ADR's revenues were $17 million—an increase of 25 percent over 1976—and for the first time, ADR declared a cash dividend on our stock, now selling in the $6 to $9 range. ADR's software products now accounted for more than 80 percent of ADR's revenues and growing with the industry at 32 percent per year.

Meanwhile, IBM was mired in the IBM/DOJ suit and was, for the most part, on its best behavior. However, it was losing market share in the database management system (DBMS) marketplace, and companies like Cullinane, Software AG, and Cincom were rapidly growing, taking market share away from IBM's

Information Management System (IMS).

In 1978, the Wall Street analysts were closely following ADR's stock and looking for even faster growth (overall, the industry was still growing between 30 to 35 percent annually). They questioned why ADR wasn't in the DBMS market, which was growing rapidly. Many companies could not afford the significant resources that IMS required and they recognized the shortcomings of the Indexed Sequential Access Method (ISAM), a restricted way of storing data. So the DBMS craze was on, and with a little push from John Bennett, I supported the acquisition of a small DBMS company from Insyte Corporation in Dallas, Texas, that offered a set of DBMS products called Datacom.

The Datacom products were excellent, and our technical people were enamored of the DBMS products. It was a state-of-the-art relational database with an active data dictionary. I believed, however, that ADR was entering the DBMS game late and I didn't want ADR to compete with the independent DBMS software vendors, let alone IBM. But I put my director's hat on and voted along with our other directors to acquire this small company with revenues of about $2 million.

We believed that ADR could integrate some of our Princeton products with the DBMS products we were acquiring. This would give ADR a unique selling proposition, and on paper, it looked quite appealing. Just prior to the Datacom acquisition, I had personally hired Adam Rin to research and develop a higher-level language to be built atop MetaCobol. Cobol, now about 20 years old, was the most widely used language for building commercial applications, so we thought, why not improve on it with additional high-level statements?

I also had just coauthored *High-Level Cobol Programming*[6] with Jerry Weinberg, Steve Wright, and MetaCobol product manager Dick Kauffman. The book gave examples of how Cobol could be raised to a higher-level language through the use of a preprocessor such as MetaCobol. Weinberg was the book's major contributor and a well-known lecturer, author, and college professor. Our shared belief in the need for a more powerful Cobol gave me added confidence that it was doable with MetaCobol.

Rin was teaching at the University of Michigan and had written his thesis on higher-level languages. I believed that the world was ready for the next-generation language and that Rin could fulfill that dream for ADR and me. He was the ideal candidate, and for his first year he reported directly to me.

Shortly after the Datacom acquisition, Rin proposed that we build a high-level language system, with the Datacom DBMS and its active data dictionary as its underpinning. MetaCobol, he felt, was not strong enough to support the higher-level language that he envisioned, but Datacom with its relational features would be. Although disappointed that he did not choose MetaCobol, I was delighted in his desire to integrate the Datacom products.

The Datacom acquisition changed ADR and how our medium-sized organization operated. Although I was a hands-on manager, I delegated the direct responsibility for the Datacom products to Bob Wickenden. But we had to do much more product planning with the intensely competitive Datacom products. The acquisition changed how we marketed our products and how we had to plan for the integration of our products. Previously, we had sold our products to mid-level IT managers. DBMSs were sold at the highest IT management levels and often involved top corporate management as well. To sell DBMSs, we had to alter our sales organization, our product planning, and virtually our entire company.

By 1978, ADR's personnel and revenues were expanding rapidly. We decided to build a new 45,000-square-foot building on a 40-acre site outside of Princeton and occupied it in late 1979. This set off a building expansion that would last well into the mid-1980s.

In late 1978, after my annual physical exam, my internist informed me I had suffered a silent heart attack (no pain and no knowledge that my heart had been damaged) earlier that year. On my new cardiologist's advice, I was restricted to doubles tennis, but very little else changed, and I went about my daily life.

Overall, the 1970s was a great period for me. I loved my work almost as much as I loved my wife. Who could ask for more? I was still able to get my arms around the company. I was in a growing industry. I had a good working relationship with John Bennett. I could still find time to write articles about software tools and software protection.

ADR also ended the 1970s on a high note, celebrating its 20th anniversary (see Figure 2, next page). Meanwhile, the DOJ, ADAPSO, and the industry all carefully watched IBM.

Quest to protect software continues: 1975–1980

During the mid-1970s, ISVs worried about the patentability of software processes and about whether the copyright law protected against the unauthorized copying of computer programs.

In 1975, the US Congress established the National Commission on New Technological

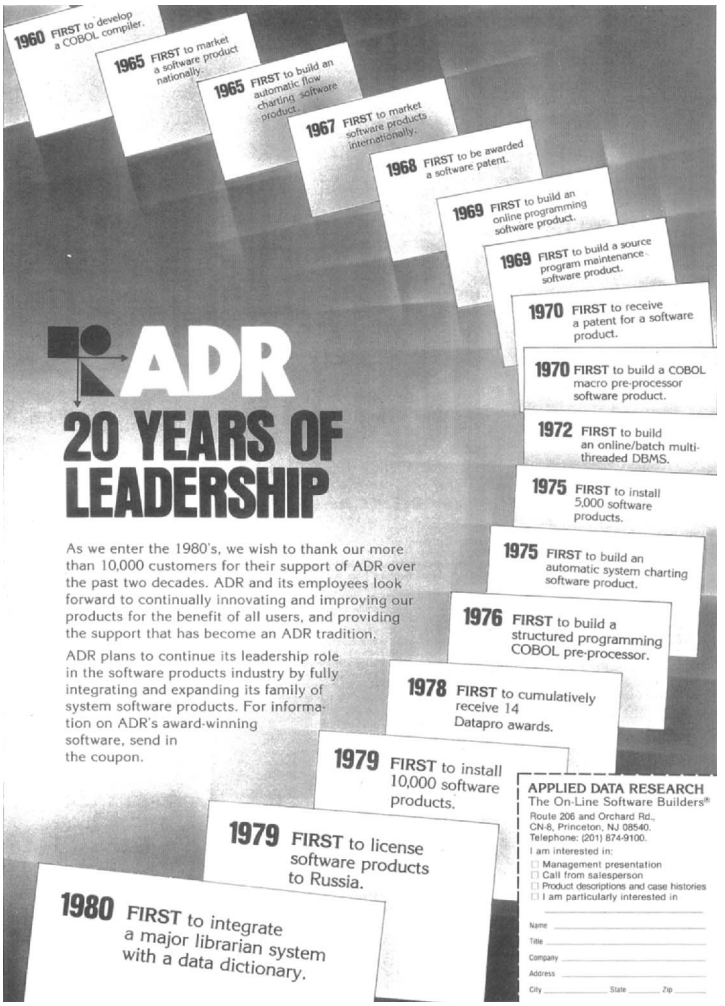


Figure 2. ADR—20 Years of Leadership. (Courtesy of the author.)

Uses of Copyright Works (CONTU) as part of its comprehensive effort to revise the copyright laws of the US. The commission held 21 meetings over two and a half years. In May 1976, ADR's counsel, Carol Cohen, represented ADR at a New York hearing, and I represented ADR at a September 1977 meeting in Chicago. As chairman of ADAPSO's Software Protection Committee, I also helped ADAPSO prepare its official position. ADR's position was consistent with ADAPSO's. We all wanted the copyright law revised to state explicitly that computer programs could be copyrighted. But we also wanted to continue to patent inventive software processes and to use a state's trade secret laws to keep computer programs and related materials secret. The commission recommended that the copyright law be amended to make it explicit that the Copyright Act of 1976 covered computer programs, and in 1980 the commission's recommendations were enacted into law.

There was little controversy over the copyrighting of computer programs, and this was one of the few times I was in sync with IBM. But controversy continued on two issues: whether software was patentable under the current patent law, and whether patenting was good or bad for the software industry.

I believed it was good for both the industry and ADR for several reasons. Software product companies were in a highly technical industry and needed the same kind of respect and protection as hardware companies. I viewed software as a tangible asset with a long life. In my articles, I wrote that a software product's life span was 20 years or more. (Later, I'd write 30 years or more.) As an industry, we were only five years old and still fighting IBM's 1960s' position that software was a service, in the public domain, and free. So a software company's ability to patent software would not only provide protection for the software invention but also raise the image of software companies, including ADR.

As Software Protection Committee chairman, I recommended that ADAPSO file amicus briefs supporting the patentability of software processes in two additional Supreme Court cases, one in 1978 and another in 1980. In 1978, the Flook patent case[7] was argued before the Supreme Court and again the Court rendered a narrow decision rejecting this patent. But this patent in no way reflected the kinds of patents software companies would file for. Both the ADR and ADAPSO briefs did not address whether the Flook patent should be issued. Instead, they argued that software companies needed to protect their software inventions. It was also argued in these briefs that a machine process implemented in hardware is equally patentable if implemented in software.

The Court ruled against Flook, stating that formulas, principles, or laws of nature are not patentable subject matter. However, Justice Stevens, who wrote the majority opinion, stated: "Yet it is equally clear that a process is not unpatentable simply because it contains a law of nature or a mathematical algorithm." Further on he cited a previous case: "While a scientific truth, or the mathematical expression of it, is not a patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be."[7] This was the third time the question of software patentability was before the Supreme Court, and again, it was left up to the industry, the lawyers, and the Patent Office to interpret the decision.

In 1980, the *Diamond* vs. *Diehr* patent case, which was a computerized industrial process for curing rubber, came before the Supreme

Court.[8] This patent was also not representative of the types software companies were trying to obtain. However, ADR and ADAPSO again filed amicus briefs on the basis that the Court would address the general question of software process patentability.

The Supreme Court opinion—favorable to the software industry—explicitly stated that "processes" were patentable, and that just because an invention used a formula, program, or computer, it was not necessarily unpatentable. The decision affirmed the CCPA's ruling and rejected the Patent Office's position, which had appealed the case. The Supreme Court did not rule on the question of whether the industrial process was "novel" and "non-obvious," which it left to the Patent Office to decide.

The Patent Office shortly thereafter came out with specific guidelines for software patents. The new guidelines were called "PTO Guidelines on Computer Inventions" and the opening paragraph said

> The US Supreme Court decision in *Diamond* v. *Diehr*. º significantly affects an examiner's analysis under 35 U.S.C. 101 of patent applications involving mathematical equations, mathematical algorithms and computer programs.[9]

The patenting of "machine processes implemented in software" was no longer a question mark for the software industry.

But copyright and patent issues continued to plague the industry as companies wanted to protect a program's look and feel as well as patent the business methods they were implementing in their programs. I continued to write articles and speak out on these subjects that today are still being debated and still largely unresolved.[10]

ADAPSO/IBM issues accelerate: 1980–1986

Many ISVs, along with ADR, were still complaining in the early 1980s about IBM's business practices and policies. The issues of availability and timeliness of interface information, of tying firmware to software, of bundling selected products with its small operating system, and of the availability of source code kept IBM on the defensive at ADAPSO.

In 1980, at ADAPSO's World Congress in San Francisco, I chaired a session called "IBM Dominance" and delivered a paper: "Why IBM Should Be Required to Have a Separate Software Company." One of the speakers was Wessel, ADAPSO's general counsel, who over the years had educated me about the merits of "maximum separation." I started my talk by quoting from a speech Wessel gave in 1975 concerning the IBM/Telex antitrust case:

> The use of economic power in one distinct line of commerce, for competitive advantage in a second, not justified by scale or other similar benefits, is anti-competitive and may constitute an unlawful restraining of trade. Where appropriate, the court should apply the principle of maximum separation to prevent such an abuse of economic power.[11]

I spoke about IBM's growing dominance in software and how a separate IBM software company would provide such maximum separation and be in the best interest of IBM hardware users. I discussed how it would reduce, if not eliminate, IBM's account control for software purchases as well as foster competition and innovation. I had hoped to influence the Justice Department, the ISVs, and IBM users who had mixed feelings about the benefits of such a breakup (ironically, this same ambivalence exists today in the Microsoft/DOJ case).

At about the same time in 1980, ADAPSO's Vendor Relations Committee became active for the specific purpose of providing a forum for ISVs to meet with the IBM representative to address IBM's business practices that were changing as the industry was maturing. Oscar Schachter, a longtime, respected ADAPSO member, Harvard-trained lawyer, and senior executive at Advanced Computer Techniques (ACT), a software products and services company, became its chairman. I was a member of that committee. ADR was an active participant at the meetings, and I was quick to air ADR's complaints. Although it's hard to know if these discussions actually helped ADR, years later I was told that it did make IBM reexamine its policies and its effect on software companies.

The IBM/DOJ suit, now being prosecuted under President Jimmy Carter, was more than 10 years old, and ADAPSO decided to update its 1973 position paper, "IBM's Monopolization of the Software and Services Industry." The 1973 paper was still relevant, and although IBM had never agreed to change its business practices as proposed by ADAPSO, it was, nevertheless, sensitive to its recommendations. However, ADAPSO wanted to have its position known should IBM and the DOJ settle the suit that was about to end.

One of the Vendor Relations Committee's first actions was to develop the ADAPSO position on the IBM antitrust action, adopted by the ADAPSO Boards of Directors on 29 April 1981. This position paper was specific in its

stands on bundling, pricing, interfaces, timing of announcements, and performance specifications. Although it had absolutely no legal impact on IBM, it became the basis for many of ADAPSO's discussions with IBM.

The Vendor Relations Committee met many times over the years, at both ADAPSO's semiannual meetings and at additional meetings in ACT's offices. The issues concerned IBM's business practices, and every issue was relevant to ADR, so I attended almost every meeting.

In November 1981, IBM announced a smaller version of its IBM 4331 departmental computer, its IBM 4321. Again, IBM's software pricing and its bundling hit ADR hard. For the IBM 4321, IBM bundled 14 of its program products with a small version of its VSE operating system and named it Small System Executive/VSE (SSX/VSE). One of the 14 products was ICCF, the $65-per-month competitor to ADR's Vollie. In the late 1970s, ADR had complained about the low price of the IBM product and had received little satisfaction. Bundling ICCF with the operating system was the last straw. IBM had done almost exactly the same thing with TSO and its large MVS operating system.

ADR vigorously complained to IBM directly and raised this issue at the Vendor Relations Committee. In March 1982, after several meetings, IBM made a small concession. It would let companies pay only for products they specifically ordered on the IBM tape and be required to delete the unpaid programs from the tape. *Computerworld* wrote about ADAPSO's "putting the heat on IBM and forcing IBM to unbundle." It was a shallow, minor victory but better than nothing.

In January 1982, the Reagan administration dismissed the IBM/DOJ case and gave IBM a clean bill of health. While the case had been in the courts for more than 12 years, IBM had been on its best behavior. Consequently, although the industry had its problems with IBM, the industry continued to expand, as did ADR.

ADAPSO continued to pressure IBM. In May 1983, it issued an important position paper: "Position of ADAPSO on New Policies of IBM." In it, ADAPSO attacked IBM for announcing in February 1983 that it would start withdrawing source code and other interface information that it had previously distributed with its operating systems and program products. Second, the paper castigated IBM for bundling its SSX products, which it feared would be a trend in the wrong direction. I had helped write the position paper, which had three sections. In the first section, ADAPSO reviewed its previous positions on the ways in which IBM should alter its business practices. In the second section, ADAPSO explained why IBM's refusal to disclose source code and timely interface information was an improper use of market power. The third section addressed the use of IBM's SSX/VSE as an improper tie-in that represented a trend toward bundling all IBM system software.

ADAPSO and ADR also contacted the European Economic Commission (EEC), which had an ongoing antitrust suit against IBM. Both ADR and ADAPSO software members shared the same goal: to slow down IBM. IBM, averse to bad publicity, wanted good relations with ADAPSO. Whether we really slowed them down is unclear, but at least we let them know we wouldn't passively sit still.

Many of IBM's strategies involved competition from the Japanese and American IBM plug-compatible mainframe manufacturers. A free operating system now made no sense to IBM. When it unbundled in 1970, IBM said that operating systems were part of the hardware. Now its free operating system, which could run on the IBM-compatible mainframes, and the release of the OS source code were causing IBM to lose potential revenue. So, slowly, IBM began to charge for its operating systems and started withdrawing the OS source code. One obvious result was to make it difficult for Amdahl and other competitors to keep their hardware compatible with the latest versions of IBM's operating systems that users would now purchase for use on a compatible mainframe.

The ISVs were caught in the middle. It was well known that in the late 1970s and 1980s IBM feared the Japanese who were flexing their muscles worldwide in many industries, including computers.[12] I remember many meetings, both public and private, in which IBM executives tried to assure ADAPSO, the ISVs, and ADR that IBM was our friend and didn't want to hurt us. In one sense, I believed them. But IBM was also concerned that the ISVs were growing their software businesses faster than IBM.

In 1984 and 1985, ADAPSO's IBM Interface Committee (previously the Vendor Relations Committee) met many times with IBM. During one six-month stretch in 1984, when the source code issue was the prime subject of discussion, the committee met almost monthly.

The source code issue—IBM's providing object code only (often referred to as IBM's OCO policy)—was particularly sensitive for ADR and for me. IBM had come full circle from its 1960s position that source code should be in the public domain and that software programs were a service. For the past 15-plus years, ADR, other

ISVs, and users had access to IBM's source code, which provided interface and other information without our having to depend on IBM's good graces. IBM, on the other hand, now said that source code was an important asset containing trade secrets. IBM's senior management openly stated that its fear of Amdahl and the Japanese plug-compatible manufacturers was paramount in its source-code strategy. I wrote and spoke about why it was unfair for IBM to change this long-standing business practice.[13] As an ADAPSO member, I visited both the DOJ and senior IBM executives in Armonk, New York, to try to change this new policy. It was still a burning issue with ISVs and ADR when, five years after IBM's OCO announcement, *Computerworld* contacted both IBM and me to debate this subject in its In-Depth column. We did so in its 8 February 1988 issue (see Figure 3).

IN DEPTH

Object-code only: Is IBM playing fair?

IBM's 5-year-old OCO policy protects its own assets but may threaten customers' investments

NO

MARTIN GOETZ is chief executive officer of Syllogy Corp., a systems software company in Hackensack, N.J. He was previously senior vice-president of Applied Data Research, Inc. and was one of the company's founders in 1959. He is chairman of ADAPSO's IBM Interface Committee and past president of ADAPSO's Software Industry Association.

ILLUSTRATIONS BY ALAN WITSCHONKE

By closing its software architecture through an object-code only (OCO) policy, IBM reduces the opportunities for companies developing complementary and competitive products. The source code vs. object code debate is clearly a competitive issue that will affect IBM's user base.

If IBM's OCO policy is not reversed, IBM users will face higher prices, fewer alternative suppliers, fewer complementary offerings and less innovative products.

YES

PETER SCHNEIDER is assistant general manager of development at IBM's Enterprise Systems in Harrison, N.Y. Formerly, he was assistant group executive of systems development in IBM's Information Systems and Storage Group. A graduate of the University of Wisconsin with a Ph.D. in electrical engineering, he joined IBM in June 1966.

IBM's software delivery practice has been in place for five years. If you examine what the company has done in regard to its object-code only (OCO) practice, it is merely following what others have done for a longer time and probably for the same reasons.

IBM believes that making licensed programs available in OCO form is a worthwhile practice because it provides customers with high-quality, functionally complete and easy-to-service software. OCO also enables the company to pro-

Figure 3. The 8 February 1988 *Computerworld* In-Depth debate article on IBM on the "object code only" controversy. (Reprinted with permission of *Computerworld* magazine.)

In the mid-1980s, as ADR became a more direct competitor of IBM in the DBMS area, I had difficulty separating my ADAPSO responsibilities from my ADR responsibilities. ADR's Datacom business was booming and IBM's DBMS market share was eroding. With the DOJ suit dismissed, IBM had become more aggressive against all the ISVs—especially its IMS and DB2 database system competitors.

ADR's growth accelerates: 1980–1986

By 1980, ADR had reached $37 million in revenues and had 12,000 product installations worldwide, more than any other independent software company. Our Datacom database product revenues had almost doubled from the year before. Software products were now accounting for about 85 percent of the ADR revenues.

The Datacom acquisition had changed ADR in many ways. A typical Datacom product suite sale was a high-level one, involving hundreds of thousands of dollars and a long selling cycle, usually culminating in an all-day visit from the prospect to our Princeton headquarters. Our new building featured customer demonstration rooms with the latest audio-video equipment. The demonstrations would be followed by meetings, hosted by our new marketing department and attended by ADR's senior technical people.

The sales and marketing effort had become more complex because our products had. I found myself getting more involved in helping to coordinate product integration, which was necessary to stay ahead of our competition. Integration was a major development effort, and we marketed it as a new world of integrated software.

In 1980, I was still managing the four technical product development groups. To manage these four groups, plan for multiple product integration, manage the marketing and sales, and have profit and loss responsibility was more challenge than I had time or skills for. So, in 1980, I started reorganizing my division and hired Joe Farrelly, a senior technical executive, as my technical coordinator. Within a year, Farrelly was promoted to technical director, with the four group product managers reporting directly to him.

In late 1981, as ADR's software division steadily grew, I hired William (Bill) Clifford from one of the major consulting firms to help me in planning and running the division. We needed consultants to assist our Datacom users in building applications, and we had to build education centers around the country for cus-

tomer training. Additionally, we expanded our data center so that we had a US and European network of computers for internal communication, demonstrations, and program development. We also needed more space. Over several years we extended our building from 45,000 to 90,000, and then to 145,000 square feet.

In 1982, ADR and the other database companies proliferated and took market share away from IBM's IMS system. IBM had just introduced DB2, its new relational database system, but was marketing it as a complement to IMS, not as a replacement. Companies found IMS difficult to program, and it required lots of computer resources. Introduced as an easy way to query a database, DB2 was deemed too slow for database updating, so DB2's introduction did little to slow the sale of ADR's or other ISVs' database products. Additionally, ADR's Interactive Development Environment for an Application's Life-cycle (Ideal) product, a fourth-generation programming language several years in development, had just been released. The Ideal product, which was integrated with Datacom's database and Datadictionary, was enthusiastically received by users. We were still technically well ahead of IBM and our other competitors.

IBM had also just introduced its personal computer, and there was great interest in being able to download data to a PC, the new client–server wave of computing. ADR's development effort was moving along in that direction at a hefty pace as all the database companies were beginning to retrofit their products to the client–server world. ADR kept pace with its competitors but at a high price. Our cash flow became negative because of our fast growth and high product development costs. ADR had several additional secondary stock offerings and raised additional capital.

By 1983, ADR's revenues had grown to about $90 million, and my division was growing by more than 40 percent annually. I gave Bill Clifford responsibility for education, consulting services, the data center, and long-range planning. Both Clifford and Farrelly became indispensable to ADR as ADR continued to grow its business.

I had now been at ADR for 24 years. We were a much larger company, and I had learned how to delegate so that I could spend my time productively, without getting involved in every problem the company faced each day. My health also remained good. I exercised regularly, played doubles tennis occasionally, and saw my cardiologist every six months. On one of my visits, I again quizzed him about whether I could play singles. This time he suggested double bypass surgery. This, he said, would eliminate a potential artery problem and also allow me to play singles. I quickly agreed, and within a month I found myself in Houston, being operated on by Michael DeBakey and his team. The operation went smoothly and within six weeks I was back at work full time and playing singles again. Unfortunately, when I was examined one year later, I found out that the two bypasses had closed, and I was back to playing doubles. But I still had no symptoms and I was feeling no effects from the closed bypasses.

ADR marked its 25th anniversary in 1984. We promoted our stability and longevity, and thought the sky was the limit. CEO and President Bennett asked me to become ADR's president and chief operating officer, which meant responsibility for all divisions, not just SPD. My promotion didn't change my focus on SPD, which now represented more than 90 percent of ADR's revenues. I quickly delegated the responsibility of the other divisions to Clifford, so my responsibilities didn't change too much.

That year we moved from the American Stock Exchange to the New York Stock Exchange. Starting in the late 1970s, the ADR stock price began to move, and we had a two-for-one split in 1983. When we went on the New York Stock Exchange in September 1984, our stock began trading at $25.

1984 was one of ADR's best years, as we increased our revenues by 44 percent to $128 million. Our Datacom products now represented almost 50 percent of the company's revenues, and our products included software for database management, application development, online programming, office automation, performance measurement, and PCs. We had more than 1,500 employees, 9,000 clients, and 20,000 product installations. ADR was now one of the world's five largest independent software companies.

Going into 1985, we were more confident than ever. That spring, we had an additional secondary stock offering at $32 per share to raise money for our continued expansion. We had budgeted for about a 30 percent growth, but this was not to be. IBM changed its database strategy and was now marketing DB2 as a replacement for IMS. It was offering six-month DB2 free trials along with free consulting services to assist companies building DB2 applications. This strategy delayed many Datacom sales, and we were well below budget in the first three quarters of 1985.

At about that time, Ameritech (the Midwestern part of the Bell system that was broken up by the DOJ), approached ADR to discuss a possible acquisition. We had been

approached by other companies before but had had no interest in being acquired. This time, it was different. Ameritech was looking to diversify and, at the same time, get into the software business. Its management planned to operate ADR as a wholly owned subsidiary. Ameritech also had a large cash reserve and was willing to fund ADR's potential growth.

At that time our stock dropped from $32 per share to about $20 per share. In the fourth quarter of 1985, the acquisition was agreed to at $32 per share (which came to about $218 million as the selling price), and early in the first quarter of 1986, the acquisition was consummated. ADR's senior executives signed two-year employment agreements, and it was business as usual.

Although ADR's revenues in 1985 increased only 15 percent—well below budget—Ameritech agreed to a healthy 1986 budget. We forecast ADR's SPD to grow by about 30 percent and continued to add personnel.

ADR and the industry typically forecast most of their revenues and profits for the second half of the year. Although we were not meeting our forecasts in the first half of 1986, we were still optimistic that we would have a good year. But IBM was putting a great deal of development and marketing dollars into DB2, and the sales cycle for database systems was getting longer.

In October 1986, however, Bennett and I saw the handwriting on the wall. I asked him to find someone to replace me. I wasn't making my revenue projections and wasn't comfortable running such a large operation. Ameritech brought in Denny Strigl, a longtime Ameritech executive, to replace me. Clifford became COO reporting directly to Strigl, and I became chief technical officer in a staff position. As CTO, I could spend more time with the development staff, write articles, and spend more time on ADAPSO issues that directly affected ADR. I could have broken my two-year contract with Ameritech, but I still loved ADR and wanted to do whatever I could to make it succeed.

By the end of 1986, IBM's repositioning of DB2 as a strategic broad-based production-level DBMS was starting to pay off. The 1 March 1987 *Datamation* published an article titled "DB2: Dressed for Success." The article stated that IBM hit rock bottom in 1984 with only 20 percent of new DBMS sales, but doubled that to 40 percent by 1986. *Datamation* attributed IBM's success to its 1,000 beta sites in 1984 and wrote, "IBM had miraculously staged one of the most remarkable product turnarounds in corporate history." I quickly responded with a letter to the editor in which I quoted *Datamation*'s statements on IBM's unusual marketing tactics that I believed to be unfair business practices.

In April 1987, IBM aggressively pushed DB2 when it announced the OS/2 Extended Edition for the IBM PC. IBM and Microsoft were jointly building a standard version of OS/2, but this special version, which contained an embedded (and bundled) DB2-compatible database management system called Database Manager, would be developed by IBM and marketed only by IBM to its mainframe users. The Database Manager would communicate with its IBM DB2 database management system on IBM's mainframes through another bundled program called Communications Manager. All the database companies, including ADR, viewed these IBM plans as threatening and illegal. Even more damaging was that IBM gave no availability date in its April 1987 announcement. Instead, it stated, "The general availability date of OS/2 Extended Edition, V1.1 will be announced in the fourth quarter of 1987."[14]

At about the same time, IBM also announced that it was building a Repository System, which was of considerable interest to most IBM mainframe installations, but that it would require DB2 as a prerequisite and would not be a standalone product. Again, IBM gave no availability date for this planned repository—which they never delivered.

ADR, ADAPSO, and IBM: 1987–1988

Having surrendered the ADR presidency, I now had more time to devote to fighting IBM. In 1987, I agreed to become ADAPSO's chairman of the IBM Interface Committee. In this role, I called for a meeting that was attended by virtually all of IBM's DB2 competitors.

Within two months, and with the committee's input, ADAPSO on 29 September 1987 came out with a scathing position paper lambasting IBM for what the ISVs viewed as an alarming trend toward bundling, preannouncements, and unfair business practices. The paper was titled "Position of ADAPSO on Three New Policies of IBM." It reiterated the 1973, 1981, and 1983 position papers on a set of business practices and operational guidelines that ADAPSO was asking IBM to follow. The paper stated that IBM's policies of bundling the SSX/VSE operating system products, its OS/2 Extended Edition bundling announcement, its new restrictions on the distribution of source code, its pre-announcement policies, and other practices were unfair. The paper recommended why and how IBM should change its policies.

I received the strong support of Jay

Goldberg, who was then ADAPSO chairman and the CEO of a services and products company. Goldberg chastised IBM at a public meeting for agreeing to provide source code (as part of an arbitration agreement) to Fujitsu Limited, a Japanese hardware company, while denying this same source code to ISVs. He likened the numerous meetings with IBM as window dressing and threatened ADAPSO litigation against IBM. He stated that ADAPSO's Boards of Directors had instructed the association to begin informal meetings with government agencies including congressional committees, the Justice Department, the Federal Trade Commission, and the European Economic Commission (EEC).

In fact, ADAPSO met with all those groups and then some. On 16 December 1987, I went with several representatives from ADAPSO to meet with E. Colin Overbury, the EEC director for competition, and five members of his staff. Five days later, ADAPSO members and I met with the assistant director of the Bureau of Competition for the Federal Trade Commission and his staff and with Michael Boudin, assistant attorney general for regulatory affairs in the DOJ, and his staff. At these meetings we reviewed the ADAPSO position paper and the effects of IBM's policies and practices on competition. The following month, in January 1988, ADAPSO's counsels, Milt Wessel and Ron Palenski, met with members from the US House of Representatives Subcommittee on Monopolies and Commercial Law, and with members of the Senate Subcommittee on Antitrust, Monopolies, and Business Rights.

These meetings did not directly change most of IBM's policies. But years later an IBM executive told me that ADAPSO's actions certainly sensitized IBM to our problems and indirectly had a real impact on how it went about its business. About a year after this series of meetings, IBM dropped the OS/2 Extended Edition and it eventually stopped marketing the OS/2 Standard Edition. It also set up a system whereby ISVs could get access to selected source code under a restricted set of conditions. ADAPSO and IBM officially declared an OCO truce in August 1988.

Post-ADR: 1988 to the present

In February 1988, almost two years after the Ameritech acquisition, I left ADR to be the new CEO of a small services and products firm in northern New Jersey, optimistic that this represented a chance to build another great company. The ex-CEO of this company had been a close personal friend for many years, and I had worked with him in my Sperry days. We quickly raised venture capital, and I also made a sizable investment in the hopes of building another software products company.

Unfortunately, my optimism quickly turned to pessimism. As I quickly discovered, business-wise my friend (now an ex-friend) and I were completely incompatible. It was an unhappy experience. Within a year, I left for greener pastures.

In 1988, ADR was sold by Ameritech to Computer Associates. In 1986, 1987, and into 1988, ADR's revenues did not grow more than 15 to 20 percent per year, well below its forecasts, and IBM continued to regain its DBMS market share. Ameritech was also recognizing that there was little synergy between Ameritech's core business and ADR. So, in late 1988, ADR ceased to exist as a separate entity, and its products and people became part of Computer Associates.

In the summer of 1988, I received a call from Tommy Spain, a former ADAPSO member who had retired from IBM and was on the board of directors of Infomart in Dallas, Texas. Infomart is a large information technology conference center with permanent office space for computer companies to demonstrate and market their products. In existence since 1985, it had created an Information Processing Hall of Fame to which several computer industry people were elected annually. Spain called to tell me they had nominated Bill Gates and me for election to its Hall of Fame for 1989. It certainly made my day. Years later I heard that Spain himself had nominated me for this award, and that was particularly gratifying.

Then, in 1989, I started a new career as a private investor by contacting venture capital (VC) firms that were investing in software firms. I decided to try to coinvest with them and to try to be an advisor to any firm that I would invest in. In April 1990, I met Bob Gailus, a partner in Schroeder Ventures who introduced me to a small firm called Knowledge Based Technologies (KBT), which had developed a state-of-the-art fuzzy logic expert system. Expert systems are a form of artificial intelligence, and at that time it was considerably in vogue with VCs. Traditional expert systems use "crisp" rules and often consist of thousands of rules. Fuzzy logic rules have the potential to replace the number of crisp rules in an application by a factor of 100 to 1 or more.

Within a short time, I coinvested with Schroeder Ventures and several other private investors in KBT, which consisted of two founders who had developed the technology. During the next three years, the VC and the

private investors continued to fund the two founders but could never raise enough money to expand.

At the time of the investment, KBT had a joint marketing agreement with IBM. Although IBM had its own expert system product, it was not well accepted in the marketplace. IBM's insurance company customers, in particular, were licensing competitors' expert systems. KBT, IBM, and I worked closely together, and for the first time in my career I was not an IBM adversary. During the next several years, KBT and IBM developed a health care fraud detection system, built using KBT's fuzzy logic expert system as the underlying technology. With health care costs increasing significantly in the early 1990s, and with health care fraud on the rise, this application became of real interest to IBM. By that time, I'd lost any animosity toward IBM, and I was happy to be partners with it.

In September 1993, IBM acquired the KBT assets, which consisted of the expert system and the fraud application fuzzy-logic rules. IBM paid KBT a small initial payment and agreed to pay royalties on the fraud detection application revenues for seven years. As part of the KBT/IBM agreement, one of the founders joined IBM as a consultant to help maintain and market the fraud detection application. At that time, IBM was losing money, in the process of laying off 60,000 of its employees, and Lou Gershner had just joined IBM to stop the drain. The outside investors, including me, were satisfied with the agreement, and we were in no position to negotiate the terms in any significant way. I went off to consult and invest in other companies and hoped over time to recover my investment in KBT.

About two years later, in August 1995, IBM decided to terminate the KBT royalty but continued to market the application. The stated reason was that IBM's internal fraud detection development group had replaced the KBT fuzzy logic technology, and under its interpretation of the KBT/IBM agreement terms, it had the right to terminate the KBT royalties. We disagreed. For about 18 months, our lawyers tried to resolve the disagreement. I personally wrote to Gershner several times asking him and his office to try to resolve this issue. The last thing I wanted was to go to court against IBM on a contract dispute. I had thought my days of fighting IBM were over. But by that time, I had personally invested a significant amount of money, and I was not about to walk away empty-handed.

Our small New York law firm had tried hard, but couldn't get IBM to move. My youngest daughter, Karen, who was a Harvard-trained lawyer, had advised me that we had probably talked too long with IBM and needed to take another approach. I was the prime contact with our law firm and was by now KBT's largest investor. To make a long story short, in February 1997, KBT finally brought suit against IBM. Shortly after KBT filed its complaint, KBT switched law firms to my daughter's law firm. Karen, who at that time was a six-year associate, took charge of the KBT/IBM suit along with a full partner from her firm. After exchanging discovery, depositions, and expert reports, in March 1998, one month before the trial was to begin, we settled out of court. KBT and I were pleased with the settlement. I felt KBT and I had been vindicated. I recovered all my investments in KBT and in the lawsuit, and then some.

In the 1990s, along with my involvement with KBT, I invested in several other software and Internet companies and often became an advisor to their senior management. I also continued to write about the need for better programming tools and, in particular, on the need for higher-level languages. I also wrote many articles critical of object-oriented programming (OOP)[15] and its theoretical reuse. Additionally, I wrote about Microsoft's business practices, which I believed were illegal and similar to IBM's past practices years ago.[16] I also continued my involvement with ADAPSO until about 1995.

In 1997, I found out I had prostate cancer and within two months had a successful operation. Today, five years later, I have no signs of a recurrence. It did help me put my priorities in order. As I was moving up in my years, it was also still gratifying to be recognized by my peers. In September 1997, the New Jersey Technology Council had a "High Tech Hero" breakfast for me that was attended by about 75 ADR alumni. And in 2000, I was inducted into the New Jersey Inventors Hall of Fame for receiving the first software patent.

Today, I enjoy being an active "business angel"—a term for private investors who invest in areas of their expertise and advise several com-

For More Information

Many of the articles, presentations, documents, and ADAPSO documents referred to here are currently archived at the Charles Babbage Institute at the University of Minnesota. See Martin A. Goetz Papers, http://www.cbi.umn.edu/collections/inv/cbi00159.html, and Applied Data Research Software Products Division Records, http://www.cbi.umn.edu/collections/inv.cbi00154.html.

Figure 4. A recent family photo: wife Norma, children Ruth and Karen, grandchildren Ben and Sarah, and me. (Courtesy of the author.)

panies in which I have investments. My health is good, and I still feel like I'm on my honeymoon after 30 years of marriage (see Figure 4).

Epilogue

Today, in 2002, the question, What is software? still has many legal and social ramifications. Proponents of the open source code movement believe that software should be put in the public domain and maintained on a voluntary basis by volunteers around the world. That, to me, is not how one develops state-of-the-art advanced software. Historically—and into the future—almost all advanced software has come from ISVs.

IBM is still one of the most powerful companies in the computer field, but little is written today about its being a monopoly (although it still has a monopoly on the operating systems for its mainframes). Today, it's all about Microsoft and what should, or should not, happen to it.

The ISVs of today are battling Microsoft. I do not envy my software brethren. IBM in the 1970s and 1980s had a hardware mentality, and in that sense, it was easier for the ISVs to battle. As an investor, I (and most investors and VCs) shy away from investing in a company that shows Microsoft as a competitor. Microsoft has a software mentality and is many more times dangerous than IBM ever was. But, as I learned, there is strength in numbers, and trade associations and the government can effectively fight large organizations that break the monopoly laws.

In these memoirs I have recounted a 20-year period in which ADAPSO developed position papers and had open and candid dialogues with IBM on important unfair competitive-practices issues. Those are the same issues that ISVs have today as they battle Microsoft and other giant software and hardware companies. In a nutshell, there are four major issues:

- Bundling—It can be an illegal tie-in if the bundler has a monopoly in one of the two markets that make up the bundle;
- Interface information—This is needed by ISVs for integration and interoperability;
- Pre-announcements—These can unfairly freeze the market; and
- Maximum separation—There should be separate companies or the like so that competition is fair.

Today's Justice Departments, whether state or federal, should review the ADAPSO history. What's at stake is a level playing field so that ISVs can innovate, survive, and prosper. With the software industry only slightly over 30 years old, there is still a great need for new and better software. It is incumbent on our government to see that it happens. Only in a robust competitive environment will innovation occur. That is why I fought so hard for ADR and the other ISVs to survive and prosper. Now it is up to today's and tomorrow's ISVs.

Acknowledgments

My sincere thanks to Mary Lou Roberts and my wife Norma for critiquing and editing these memoirs.

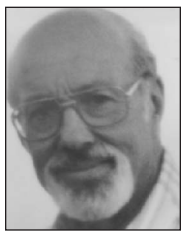
References and notes

1. The term *account control* as applied to IBM had a special meaning. IBM sold its computer hardware at a company's highest executive levels. It tried to use the relationships formed at these levels to influence and control internal decisions made at middle management levels. An example of IBM account control can be found on the Software History Center Web site (http://www.softwarehistory.org/history/Kolence1.html) as "How IBM Killed the Market for Boole and Babbage's CUE Product."
2. The 1956 Consent Decree required that IBM operate its Service Bureau Corporation (SBC) as a completely separate entity and not in any way capitalize on IBM's name, employees, reputation, and financial assets. The entire Consent Decree, which describes how IBM and SBC was enjoined and restrained, can be found on the Internet at http://www.cptech.org/at/ibm/ibm1956cd.html.
3. The Prater and Wei patent, originally filed by Mobil Oil Corporation in 1960, covered a computerized spectrographic analysis technique. It

had many method and apparatus claims that could be performed either on an analog or digital computer or with pencil and paper.

The Prater and Wei decision on 14 August 1969, by the Court of Customs and Patent Appeals (CCPA), is famous because the question "whether computer programs could contain patentable subject matter" was also before the CCPA.

4. Brian Randell, who served as editor for these reports, has put them on the Web: http://www.cs.ncl.ac.uk/people/brian.randell/home.formal/NATO/NATOReports/.
5. The Johnston patent involved a machine system for the automatic record keeping of bank checks and deposits. It was a business method that bankers had always used on paper previously. While the world hoped for a landmark broad decision by the Supreme Court on software's patentability, it made a narrow decision stating that the patent did not meet the test of "unobviousness" and rejected the patent claims.
6. M. Goetz, R. Kauffman, J. Weinberg, and S. Wright, *High Level Cobol Programming,* Winthrop Publications, Cambridge, Mass., Feb. 1977.
7. The Flook patent involved a formula for computing variables while continuously monitoring the results, so that an alarm would be set if a predetermined limit was reached. The Supreme Court's review of this patent was its third review in six years of a patent case involving software-related issues (the two previous cases being *Gottschalk* vs. *Benson* and *Dann* vs. *Johnston*).
8. The *Diamond* vs. *Diehr* patent case involved a computerized process for curing rubber, by determining the length of time a particular batch of rubber must remain in a molding press in order to be perfectly cured. The Supreme Court in a 5-to-4 decision ruled that an industrial process could be eligible for a patent even if it incorporated a mathematical formula or computer program.
9. "PTO Guidelines on Computer Inventions," Bureau of National Affairs, Washington, D.C., Fall 1981.
10. Some of the articles I wrote and speeches I gave covered the *Lotus* vs. *Borland* "look and feel" copyright issue and questions that were still being asked about software's patentability.
11. M.R. Wessel, "The Real Meaning of Telex," speech, ADAPSO conference, 17 Apr. 1975.
12. By the late 1970s, it was well established that Japan was gaining market share for mainframe computers similar to its increasing market share in computer components. In *IBM and the Data Processing Industry* (F. Fisher, J. McKie, and R. Manche, Prager, New York, 1983), p. 430, the authors write about Japan's interests in the late 1970s and early 80s: "The Japanese clearly intended to acquire a significant share of American computer markets, perhaps similar to their share of the US automobile market (18-20 percent). By 1980 they had captured an estimated 42 percent of the American memory market."
13. Since 1983, when IBM announced its plan to withdraw source code over a 10-year period, in public presentations, articles, and through ADAPSO, I vehemently expressed my belief that IBM's withdrawal of source code would reduce an ISV's ability to compete against IBM. When in September 1987 IBM made its source code agreement with Fujitsu, the ISPs were even more upset that IBM was selling access to its source code to a Japanese company and freezing out the ISVs. This agreement triggered ADAPSO to take aggressive actions that are discussed later on in my memoirs.
14. IBM programming announcement, "IBM OS/2 Extended Edition, V1.1," IBM Information Systems Group, Rye Brook, N.Y., 2 Apr. 1987, p. 1.
15. From 1992 though 1997, I published six articles on why OOP was not a practical method for developing enterprise applications in MIS organizations and that the ability to discover reusable code through OOP was largely a myth.
16. In 1995, Microsoft had just concluded a consent decree with the US DOJ and there was little belief then that such an agreement would curtail Microsoft. I wrote a series of articles at that time comparing how the software industry fought IBM in the 1970s and 1980s and that it was an effective way for ISVs to fight Microsoft.



Martin Goetz is a private investor and management consultant to software product firms and venture capital firms. He was a founder of Applied Data Research in 1959 and former president of ADR. He received the first US software patent in 1968. In 1989, he was elected to the Infomart Computer Hall of Fame and, in 2000, to the New Jersey Inventors Hall of Fame. He is recognized for his work in software product protection through copyright and patent law and for combating unfair competitive practices in software by hardware manufacturers. Currently, he is a trustee and a director of the Charles Babbage Foundation.

Readers may contact Martin Goetz at marty@goetzassociates.com.