

Memoirs of a Software Pioneer: Part 1

Martin Goetz

In the first of this two-part series, these memoirs trace the software industry's beginnings back to the early struggles by software firms such as Applied Data Research as they tried to compete against IBM, the computer industry's Goliath. The author identifies the questions raised, from the antitrust suits and related issues, that led to IBM's unbundling in 1970.

Part 1 of Martin Goetz's memoirs covers his career through 1970. Part 2, concluding his memoirs, will be published in a subsequent issue of Annals.

Some events—such as when the universe began—are hard to pin down. But I, and many others of my era, know where, when, how, and why the software industry started and how it evolved.

Today, the software industry is readily talked about in the news, on Wall Street, and in every company from the smallest mom-and-pop operation to the Fortune 500 giants. But when I started working as a computer programmer in 1954, there was only a small computer hardware industry and a few scattered software services firms. Now, not even 50 years later, the software industry has become America's third-largest manufacturing industry (behind automobiles and electronics), and it boasts annual revenues of well over \$100 billion. Although the software industry encompasses software products and services, my experience has been entirely in the software products area, so that is what these memoirs cover.

I believe that software products, as a business, began about 1964, when Applied Data Research (ADR), a company I helped found, began selling a software program called Autoflow. From 1964 through 1970, many other companies also began selling software products, but by no stretch of the imagination could it yet be called an industry. In my opinion, IBM's unbundling announcement in 1969 was what turned a nascent business into the software industry.

My definition of a software product is simple. It is a generalized off-the-shelf program, sold separately, which can be used by many users without their having to significantly modify the source code. IBM called the software they sold a "program product" when they unbundled their software. Although my definition of a software product may be controver-

sial, it is probably the least controversial of the software issues that began to be debated in the 1960s—and are as controversial today as then.

Many of the issues raging today in the industry and in our courts are the same ones I have been involved with since the mid-1960s. When is adding a component or feature to an operating system (for example, an Internet browser) an illegal tie-in? Should business method patents in the US continue to be issued by the US Patent Office? (The UK has made the decision not to issue such patents.) Should source code be in the public domain? If so, should it be free and available to everyone? (The Free Software Foundation and Open Source Initiative [OSI] groups think so.) What is software? Is it tangible or intangible? How should software be taxed under existing state and federal statutes? What are the obligations of large dominant companies (for example, IBM and Microsoft) in providing and releasing interface specifications for their software offerings? And in what time frame?

These issues concern the nature of software, who should own it, and how it should be protected. Although the Internet has been the catalyst for an examination of how to protect the music industry and the scope of copyright protection, those same kinds of questions arose in the early to mid-1960s when software companies began to market programs for large-scale computers. Even in the early 1980s, the nature of software became fuzzy as PC software companies started calling themselves "software publishers," as if software is a "writing" (as opposed to something like a machine that is built and maintained).

As a founder and executive at ADR, my career and life have been intertwined with these questions and issues. They directly affected ADR's growth and survival—always my first priority. I admit freely that I have been tenacious in presenting and defending my positions and my company. Perhaps watching my father fail to survive the Depression of the 1930s instilled



Figure 1. My Univac graduating class (author is at the far right, front row). (Photo courtesy of the author.)

in me the strong need to succeed. Whatever forces drove me—this is my story.

Growing up: 1930–1954

I was born the second of two children on 22 April 1930, shortly after the beginning of the Great Depression. I sometimes wonder if I heard the crash in October 1929, when my mother was carrying me. At the time, my father had a haberdashery store, where my mother helped out. Until about 1934, we lived in a middle-class neighborhood in Brooklyn, New York. But in 1935, my father lost his business and was reduced to peddling ties on the streets of Manhattan. For the next seven years he struggled to eke out a living. Times for my family got tougher and tougher as the Depression continued.

From the age of five through 12, we moved six times, each time to a lower-priced apartment. My father couldn't support his family, and my mother worked. For a short time, we were on welfare (which then was called "being on relief").

I always felt secure and loved. Yet, as is true for many others of my generation, those Depression days affected my entire life. Even today, I am financially conservative, believing that a depression could happen again, wiping out my family's entire savings. At the age of 12, on my own initiative, I started working after school delivering meat for a local butcher for 5 cents per delivery. From that point on, I continued to work part-time jobs through public school, high school, and college. Each summer, during my college years, I worked full time as a

waiter in New York's Catskill Mountains, earning pretty good money.

As a young person, I had little career guidance from my family. My father got quite sick in 1940 and died three years later from heart failure. If anyone was my role model, it was my mother, who worked in the Brooklyn Navy Yard during most of World War II while raising my brother and me. (She remarried in 1947 and lived and enjoyed her life to the fullest, passing away in 1999 at the age of 97.)

In 1943, I passed a special New York City exam that permitted me to attend Brooklyn Technical High School, then considered one of the city's best high schools. As a pre-engineering student, I graduated in 1948 with mediocre grades. Not certain that I wanted to attend college, I enlisted in the US Army (under a one-

year enlistment program) and spent a year with the Second Armored Division in Killeen, Texas.

Following my discharge in 1949, I decided to attend Brooklyn College. After changing my major from pre-engineering to economics to business administration, I transferred to the Baruch School of Business at City College of NY (CCNY), graduating in 1953 with a BBA in business statistics. I then continued at night for my master's and received an MBA from CCNY in 1961.

During my last year at CCNY, I worked part time in a work/study program at Dancer, Fitzgerald, and Sample, a large Manhattan advertising agency. After I graduated in 1953, they offered me a full-time job as a supervisor in the Media Research Department, analyzing Nielsen radio ratings. Not knowing what I really wanted to do (or could do), this seemed like a safe haven, so I accepted. Many of my CCNY friends had taken jobs in quality control, but that didn't seem especially exciting. I sensed I was not destined to work in statistics, but at the same time, I had no clear direction for a career.

Early years in software: 1954–1959

In October 1954, the *New York Times* carried a Remington-Rand advertisement for a programmer trainee at its Univac Division: "No experience required." Little did I know that responding to this ad would lay the groundwork for my career and my life. To my amazement, they hired me, and I soon found myself in a 12-week training course, learning how the Univac computer worked internally and how to program it (see Figure 1 for a picture of my

graduating class). We were exposed to some of Univac's rather crude programming tools and to BIOR (Business Input-Output Rerun), the monitor program that Univac was developing.

BIOR, my first lesson in computer manufacturers' promises about future software developments, was never fully completed or debugged. One user noted, "The Input-Output didn't work, and they never delivered the Rerun. The only thing they gave us was the Business."

I took to programming like a duck takes to water. It was fun, and I quickly learned about main-line programming, subroutines, building programs as a hierarchy in small modules, organizing programs for easy debugging, and making program changes. I loved it and used to program in my head as I drove my car. And, like most strong programmers, I never flow-charted; I just wrote the program.

We didn't use punched cards, even in 1954. Remington had the Unityper, an offline device that produced 80 character images, from keyboard input, on a small magnetic tape. There were crude programs for editing the punched card images. So when ADR developed the Librarian, a source program maintenance and security system (first in 1965 to use internally and then as a software product in 1967), the underlying concept was far from new.

After graduating from my training course, I was assigned to Consolidated Edison, New York City's large gas and electric utility, which was in the process of trying to convert from IBM punched-card tabulating equipment to electronic computers. I quickly learned that Con Ed was going to test the Univac I equipment in parallel with testing IBM's latest computer, the IBM 705—both companies built identical applications with a "may the best company win" attitude. It was an intense race and my first experience competing against IBM.

I became aware that Remington Rand had sued IBM for antitrust violations. The Department of Justice also had an antitrust suit against IBM for tying in their tabulating equipment with the sale of their punched cards. Little did I realize at the time that I would spend much of my career competing against IBM and getting involved in antitrust issues.

At Con Ed, I worked on a team of several Remington programmers, headed by John O'Neill, who was responsible for designing the application. Con Ed had selected 15 Con Ed personnel from its various departments and trained them to program the Univac and assist in the project. Some of their best programmers never went to college; in fact, their best programmer was previously a utility linesman who

spent his career climbing poles until he was selected to join the programming group. The lesson stayed with me my entire career. When it comes to programming, education is second to natural talent.

Several years later, after both systems were completed, I learned that Con Ed had selected IBM over Univac in a political, rather than technical, decision. That was to be the history of Remington during my four years there—good technology, but bad marketing. Because of their tabulating equipment, IBM had account control and used it effectively. Remington was no match against IBM. As I learned the hard way over the years, IBM's account control for selling their hardware was just as effective for selling their software.

Within a year of my assignment at Con Ed, I was asked to head a similar group at Parke-Davis, a pharmaceutical firm in Detroit, Michigan, which had just purchased Remington's newest computer, Univac II. I could fly home each weekend at Remington's expense, and it was an opportunity to head up an installation. Age 25 and single, I quickly agreed to go.

Remington Rand was running very fast at that time. Because most of its contracts stated that there would be no payments until a site was operational, the company emphasized selling the hardware, hiring and training programmers to go to new installations, and getting those new installations operational as fast as possible.

In 1955, Remington Rand had merged with Sperry Gyroscope to form Sperry Rand. From my perspective, nothing had changed except the name.

During my four years at Sperry, no real system programming group existed for building programming tools, but there was a research group in Philadelphia where Grace Hopper and others were doing experimental software research. In addition, programmers at Sperry and at Univac user sites had begun to donate programs to a shared library. The Univac Program Distribution Library consisted of programming tools and statistical programs that Sperry distributed free to its users. While it administered this library, Sperry took no responsibility for program or documentation integrity. IBM had a similar library of field-developed programs (FDP).

In my spare time, I became one of the library's biggest donors, especially for sort programs. It began quite by chance. Betty Holberton, a noted programmer at the Bureau of the Census and an early Univac I user, had

INTERNATIONAL BUSINESS MACHINES CORPORATION

DATA PROCESSING DIVISION
112 EAST POST ROAD WHITE PLAINS, NEW YORK PHONE WHITE PLAINS 9-1900

To: Branch Managers February 27, 1959

Subject: 7070 Automatic Programming

Reference: My letters of January 14, 1959, and December 5, 1958

I am pleased to bring you a further report on the status of our 7070 Automatic Programming. In order that you may review and make preliminary plans with your customers for programming and program testing, here are the target dates for release of the Automatic Programs:

Program	Specifications *	Program Release **
Basic Autocoder	Published	November 1, 1959
Autocoder	April 15, 1959	March 1, 1960
Input-Output Package	April 15, 1959	December 1, 1959
Basic FORTRAN	May 1, 1959	February 1, 1960
Sort and Merge	May 15, 1959	March 1, 1960
650 Assembly for 7070		May 15, 1959
704 Simulator		May 15, 1959
FORTRAN	June 1, 1959	April 1, 1960
RAMAC Programs	June 1, 1959	April 1, 1960
650 Simulator for 7070	July 1, 1959	January 1, 1960
Report Generator	July 1, 1959	May 1, 1960
Utility Routines	August 15, 1959	January 1, 1960

*Date on which material will be available to enable the customer to include that portion of the system in his planning.

**Date on which the program will be ready for customer test.

While program availability is predicated on a tight Applied Programming schedule, every effort will be made to meet or better these dates. We believe that they are sufficiently firm to enable you to be sure your customers take proper action to maintain the installation dates they have indicated.

Many customers will be ready to take advantage of these Automatic Programs in 1960, while others will not have progressed this far. This will provide an excellent opportunity to review and firm up definite installation dates, as well as give our customers an indication of the scope and realistic support of our 7070 Automatic Programming efforts.



C. Garrison, Jr.

Copy to: Marketing & Service Executives (DPD HQ)
Regional Management
District Managers
Special Representatives

Figure 2. A list of system programs IBM was building in 1959 for the IBM 7070. (Photo courtesy of the author.)

donated a sort generator that was widely used. Because the Univac I was a word machine, and records of different word lengths needed different fields to be sorted, her program generated the correct record size to be sorted and the location of the keys, based on a set of user parameters. The program was flexible but slow, and when I looked at the code, I quickly recognized why.

The Con Edison user application had many sorting requirements, and a faster sort would speed up the application significantly. Because we were always in a race with IBM, I tried during my spare time to see what I could accomplish. I programmed a 10-word sort with a single key. The result was a much faster sort because my program did not generate code. I wrote the main sorting loops using straight-line

coding so that there were few transfers or address modifications. In addition, I wrote the code minimizing the latency time inherent in the Univac's mercury delay memory. The elapsed time was significantly faster than the sort generator's. Because of this, I gained the reputation in Univac as a sorting expert, and over the next several years, I rewrote the sort programs based on requests from users for 2-, 4-, 6-, and 20-word sorts.

Later, I worked at several other installations, including the Philadelphia Navy Yard and John Hancock Insurance. All the while I was building more sorts, debugging aids, generalized I/O routines, and other programming tools that I donated to Univac's Library.

Over time, I became disillusioned with Sperry. IBM was beating them consistently, and Sperry apparently didn't recognize the importance of developing programmers' tools. So, in November 1958, I resigned from Sperry to join IBM's Applied Programming group in New York, a group dedicated to developing programming tools.

At IBM, I was assigned to build a monitor for the IBM 7070. This program would eliminate the need for separate tape-to-printer, card-to-tape, and tape-to-card peripheral equipment. But halfway through that effort, I was told to drop the assignment because IBM had announced its 1401 machine to handle those functions. Somewhat frustrated, but still enthusiastic about IBM, I joined the 7070 Autocoder team. I was busily working away when I heard that a group of Sperry programmers and several Sperry users had formed a new company, ADR, in Princeton, New Jersey.

One of these programmers was Steve Wright, a man with an outstanding record at Sperry whom I knew well and admired greatly. It didn't take me long to contact him. Within several weeks, I left IBM and began my 28-year career at ADR.

Before I discuss my long involvement with ADR and ADAPSO, the software industry's trade association, I think it important that I reflect on my work experience with Sperry and IBM.

It was clear in the 1950s that programming tools were important for building applications. Taking a more serious approach, IBM's Applied Programming department had a large group of programmers, which IBM expanded significantly over time. Figure 2 is a list of system programs IBM was building in 1959 for the IBM 7070. Similar programs were being built for IBM's other large mainframe computers, the 705 and 709.

Up into the 1960s, these programs were

freely given away, and they were not being built in a competitive environment. Certainly, by that time I had a vested interest in this issue because ADR had entered the software products business in 1964, five years before IBM unbundled. In 1966, I began to write articles about the poor state of system software that was still being provided free of charge to customers.

In May 1966, I wrote an opinion article for *Datamation* titled "Today's Commercial Software: Has Inertia Set In?," which began as follows:

From 1956–1966, the number and use of computers by industry has increased at least ten-fold. During that time, the software provided by computer manufacturers has not significantly changed in scope or concept. For the benefit of those skeptics among us, I quote from an advertisement that appeared in 1958: 'Software provided with the IBM 7070 includes an assembly system, a sort/merge, FORTRAN, an I/O package, multiprogramming and miscellaneous utility programs.'

One would expect that over a 10-year period, new software would emerge that would reduce programming costs and/or make programming easier. However, it appears that programming today is more difficult, and the manufacturer is content with providing the same software tools that he previously provided, tools which eventually will be viewed as very rudimentary.

My view of the software proposed by all computer manufacturers is that it is not enough, and it is not the right kind. The manufacturers' emphasis is now on operating systems and new languages ... but what will they really do for the average user?

The article goes on to describe and give examples of needed programming tools and generalized business packages, some of which were then being built by independent software companies such as ADR and Informatics.

Over the next 30 years, I wrote many more articles critical of the hardware manufacturers who, through their business practices, were impeding the growth of the software industry in general and of ADR in particular.

My early years with ADR: 1959–1970

In September 1959, I joined ADR as its first employee. Of the six founders listed in ADR's Letters of Incorporation, one remained a silent partner for about a year; the others were in transition from their current positions.

Ellwood (Woody) Kauffman, one of the founders, was an independent consultant and

former Univac employee. He lived near Princeton and was ADR's first president. He loved to say that he selected Princeton because it was equally inconvenient to both New York City and Philadelphia. Except for Woody, the rest of us had to relocate to Princeton. As ADR's only full-time employee, I sat waiting for the telephones to be installed in the small, empty office that Woody had selected. The founders were quick to give me an equal founder's share of the company; I paid \$3,500 for a one-seventh share in ADR.

The other founders were Steve Wright and Dave McFadden, who were in the process of leaving Univac, and Bob Wickenden and Bernie Riskin, who were leaving the C&O Railroad, a Univac site in Cleveland. Sherman Bluementhal, a former Univac employee, was at General Librascope, a large government contractor. He joined ADR about a year later.

ADR's businesses in the 1960s

General Librascope had a large contract with the Federal Aviation Agency's testing center, the National Aviation Federal Experimental Center (NAFEC) in Pleasantville, New Jersey. Sherman was influential in quickly getting a contract for ADR to build several programming systems for the custom computer General Librascope was delivering to NAFEC for aircraft conflict prediction. For the next two years, about eight of us, including some new hires, worked at NAFEC building sort–merge programs, an assembly program, a basic operating system, and some programming aids.

ADR continued for several years strictly as a programming services company, building programming tools for many of the hardware companies and government agencies that had special one-of-a-kind computers. One of our early projects in 1960 was to work with RCA in building the first Cobol compiler. Most of our contracts were fixed-price and competitively bid.

Although I quickly became a supervisor for many ADR contracts and did little programming myself, I stayed up-to-date on sorting techniques and recognized that, because sorting often consumed 50 percent or more of an application's processing time, there was an intense need for better and faster sorts. That interest remained with me for many years as I continually devised faster ways to sort on existing tape computers and on some of the newer disk storage computers.

Capitalizing on my experience with sorting, ADR secured a number of fixed-price contracts. To gain additional exposure for ADR's sorting expertise, ADR sponsored an ACM Sorting

Volume 6 / Number 5 / May, 1963

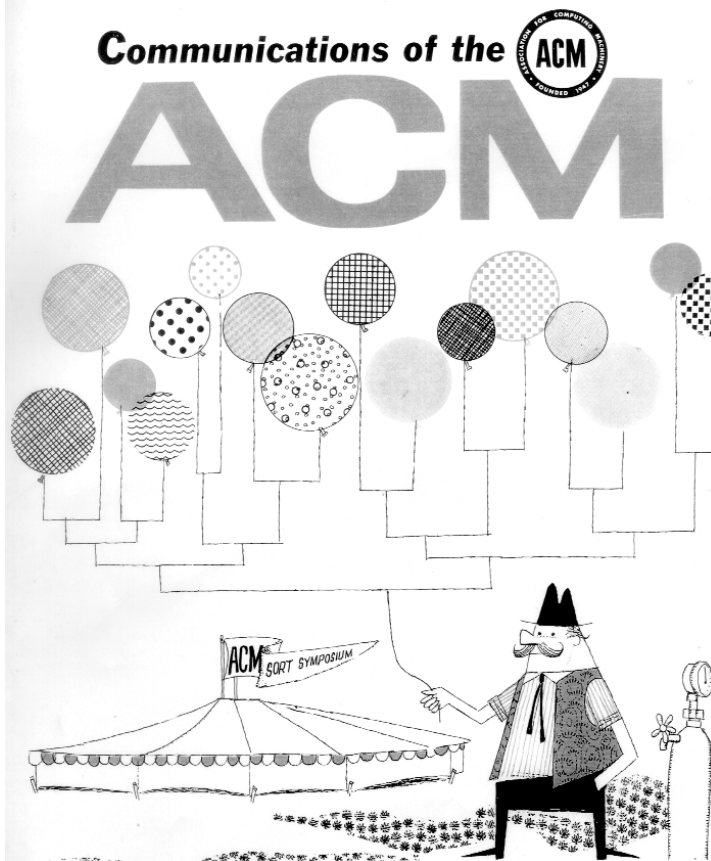


Figure 3. The cover of the May 1963 *Communications* issue devoted to the ACM Sort Symposium. (Cover illustration *Comm. ACM*, vol. 6, no. 5, May 1963, courtesy of ADR; artist Charles G. Stern.)

Symposium in November 1962. I chaired this symposium and delivered four papers on new tape- and disk-sorting techniques. Other ADR employees delivered three additional papers as did representatives from IBM, Honeywell, RCA, and various universities and government agencies. In May 1963, ACM devoted an entire issue of the *Communications of the ACM* to the topic, publishing all 16 of the presented papers (see Figure 3).

During the 1960s, ADR continued to build many sorting systems. We built a Bendix G-20 sort, several RCA sorts for the 501 and Spectra Series, a CDC 1604 sort for the Navy, a Univac LARC sort, a Sylvania 9400 sort for army intelligence, a sort on IBM's somewhat secret Harvest computer for NSA, and others.

By 1963, ADR had grown to about 30 people, and we found ourselves competing with the few other software firms that had emerged—Computer Sciences, Computer Usage, and Computer Applications. Although

we were generally profitable on each project, our cash flow was poor, and we generally just tried to survive. We did little planning, giving scant thought to patents, copyrights, trade secrets, and protecting intellectual property.

There also were some management conflicts. In 1962, Sherman Bluementhal and Bernie Riskin tried to oust Woody Kauffman from the presidency and failed. The two of them then resigned from ADR and set up a new programming services company called NCA (National Computer Analysts), which soon became one of our competitors.

Meanwhile, ADR had hired Dick Jones, also a former Univac employee, as sales manager. In late 1963, about 18 months after Dick joined the company, the founders decided that Woody was not growing the company effectively so ADR's board elected Dick Jones as the new president. Although Woody then chose to leave ADR, he remained on the board for many years and always remained a good friend of the founders and ADR.

With Dick Jones at the helm, the company began to grow much faster, and ADR opened several programming services offices around the country, including in Washington, D.C., Los Angeles, and New York City.

Dick hired a former Univac colleague, John Bennett, to run the Washington, D.C., office. John was later to become, in 1970, ADR's third president when the board ousted Dick Jones during a turbulent period for ADR, which I address later.

In spite of increasing revenues each year, ADR had little to show for it. We had little money in the bank and difficulty meeting each payroll. The need for additional investment dollars became increasingly clear when, in 1964, ADR entered the software products business with Autoflow, an automatic flowcharting program, and which we had built for the RCA 501. Although we built it on a shoestring, we clearly would be unable to market it on a shoestring.

With Dick Jones as the leader, we felt that it was time to go public, even though our 1964 revenues were less than \$1 million. Our 1965 public offering raised \$6 million, and ADR began trading over the counter at \$5 per share.

In ADR's 1965 annual report, Dick Jones noted that ADR's proprietary Autoflow product would be built for several new computers and that ADR was presently building additional proprietary products. Dick was enamored of the potential market for software products. He quickly authorized the building of a product called PDQ, a document search program, to be

built in our Washington, D.C., office. Then, in 1967, ADR's subsidiary called Massachusetts Computer Associates (which Dick had pushed to acquire) built the IAM product, which performed interactive algebraic manipulation.

Dick Jones ran ADR as a benevolent dictator. Although we had an active board of directors, including outsiders who represented the investment firm that took us public, few planning decisions were made at the board level. And, although we became a public company in 1965, ADR had no real cost controls and no profit-and-loss centers except for its two branches and its one subsidiary.

I had an excellent relationship with Dick, who was a huge supporter of proprietary products and who gave me a great deal of latitude. He quickly put me in charge of the proprietary products ADR was building in its Princeton office, as well as those built in Washington and by our subsidiary in Massachusetts.

As I reflect on my years at ADR from 1964 through 1970, it seems ADR and I fought battles on many fronts to survive, some of which were very public. For example, we used the press through articles, speeches, and advertisements to make our positions and problems known. Our first major battle was with IBM and its free product, IBM Flowcharter, which competed with Autoflow. That battle between ADR and IBM evolved into an industry battle to force IBM to unbundle. ADR had to protect its investment in the new products it was building or planned to build. Concurrently with those battles, we began to recognize the need to protect our intellectual property, and we battled for the patenting and copyrighting of software. Interestingly, IBM, a big proponent of the patent system in general and patenting computer hardware in particular, was against the patenting of software.

As a young bachelor in my 30s, I channeled my energies into those fights and in developing ADR's Software Products Division. At the same time, I was battling Dick Jones' shoot-from-the-hip approach to management. In addition to our programming services groups, Dick Jones had moved ADR into various other businesses including time-sharing, service bureaus, research, and turnkey systems—for the most part without the Board of Directors' approval and without even an annual budget or effective cost controls. As a director of the company, I certainly share some of the blame. I rationalize at this time that I was too busy doing other things. With no prior executive management experience, I was learning the hard way.

At that time, we also faced significant financial difficulties. One factor was that the Financial Accounting Standards Board (FASB) accounting rule that was in effect in the 1960s permitted software product development costs to be capitalized and amortized over a five-to-seven-year period. Therefore, many companies, including ADR, were able to show accounting profits. So, although ADR showed a book profit each year between 1965 and 1970, it had a consistently negative cash flow, and it was not financially solid.

In 1968, ADR raised additional money through another public offering. The company was now ardently followed by the Wall Street analysts as an up-and-coming company in the new and exciting software products industry. ADR's 1968 stock offering price was \$25 per share, and within a year, it peaked at \$40. In 1969, ADR had a market valuation of almost \$40 million, with annual revenues of only \$6.2 million. Like the recent Internet craze, however, sanity eventually returned to the stock market and dropped the value of ADR's stock.

I was unhappy with our negative cash flow, but I was mesmerized by ADR's market value. I continued to concentrate all my energy in making ADR's Software Products Division successful. Although ADR had actually gotten into the software products business in 1964, it was not until the 1970s and 1980s, well after IBM unbundled, that software became a major revenue producer for ADR.

ADR's software products business in the 1960s

Originally, ADR had no intention of marketing software products in a bundled environment. All software in the early 1960s was given away free by the hardware manufacturers, and users freely exchanged programs through SHARE, GUIDE, and other user groups. Users were certainly not demanding priced software.

RCA was an up-and-coming computer manufacturer at the time, and, in 1963, they approached ADR about building a flowcharting system for its users. On speculation, we designed a semiautomatic flowcharting system that accepted assembly language input and submitted the proposal to RCA. The system required a one-character code in the comments section on each line of the assembly language code to generate the appropriate flowchart symbols. But RCA showed little interest. We submitted unsolicited proposals to several other manufacturers and received the same unenthusiastic response from all.

At that time, ADR had the free nighttime use of an RCA 501 and decided to develop a

prototype to prove the flowcharting system's feasibility. A young programmer, Mike Guzik, who had just joined ADR from RCA, was assigned the task. Mike wrote the Autoflow program in four to six months. In addition to producing extremely readable flowcharts, he produced several cross-reference listings that proved helpful during debugging. We showed the completed prototype program to RCA but still found little interest.

Having completed Autoflow with perhaps a \$5,000 to \$10,000 investment, we decided to try to license it to RCA's base of about 100 RCA 501 users. I became the project manager and handled its marketing, development, pricing, and therefore its success or failure. I prepared some descriptive marketing literature and wrote to all 100 users. We priced Autoflow at \$2,400 and licensed just two copies.

It was now 1964. The IBM 360 had been announced, but there were still thousands of IBM 1401s and 1410s in use. Because the market seemed ripe for a flowcharting system, we decided to build Autoflow for the IBM 1400 series marketplace. Within 10 months, with a team of two, we produced a 1401 Autocoder version of Autoflow and quickly hit the streets, counting on the fact that many of the 1401 Autocoder programs had never been flowcharted despite its being part of the required documentation in most data processing organizations.

The interest was evident, but there was one hitch. Our system required those one-character chart codes to trigger the right flowchart box. Companies were interested in Autoflow only if the system could automatically process existing Autocoder programs without inserting the one-character codes. The project programmer quickly concluded that the operation code could indicate to Autoflow the required flowchart box type. The number of flowchart symbols would exceed those produced manually or under the control of the one-digit chart code. Nevertheless, Autoflow would produce completely accurate flowcharts that would be useful when the program had to be maintained or changed. Additionally, it filled the documentation need and satisfied corporate requirements.

Although I was now concentrating on developing and marketing the IBM 1401 Autoflow, I continued to supervise the many sorting systems that ADR was building under contract. In 1964, I divided my time between those two passions: Autoflow and sorting.

While attending the 1964 Spring Joint Computer Conference (SJCC) in Washington, D.C., to present a paper on a new sorting technique, I attended the session "Patents and Other

Legal Problems Relating to Electronic Computers," chaired by Morton (Mort) Jacobs. The panel discussed patenting software and other ways to protect intellectual property (IP). I immediately set up a meeting with Mort to discuss the possibility of patenting one of the new sorting techniques I had presented at the conference. For me, the session was a turning point, marking my introduction to Mort Jacobs and beginning 30-plus years of involvement in IP concerns.

To understand the issues we faced relative to patenting our sorting techniques requires a little knowledge of how sorting large numbers of records on computers works. Sorting theory for sorting large numbers of records on general-purpose computers with tape units was well understood in 1964. The first phase, called the *internal sort*, creates many sorted strings of records using the computer memory and writes the strings of sorted records on up to $n - 1$ tape units. The second, *n-way merge* phase continually merges the strings of records (to form longer strings) onto the various tape units until there is only one long string (consisting of all records) on one tape. At that point, the sort is completed.

The trick to reducing the elapsed time for the entire sorting process is to make the initial strings as long as possible in the first phase and to do the largest-size *n-way* tape merge in the second phase. Maximizing the "way" of the merge reduces the number of times the records are read and written, thereby significantly reducing the overall sorting time. This can also reduce the waiting time as tapes (which can be read only in a forward direction) are being rewound between successive merges.

Over the years, ADR and I had developed some interesting ways to improve these two sort phases, some of which I presented in my paper at the SJCC. Although ADR had never built any sorts for IBM computers, I was familiar with IBM's tape units that could only read and write records in a forward direction. I began to wonder if a faster sort product in competition to the free IBM sort would be feasible. ADR could take some of the techniques it had developed for other manufacturers' computer systems, as well as the new merging technique I had developed, and apply it to IBM's 360 system.

Mort Jacobs convinced me that I could get patent protection for my new sorting technique. If awarded, this might stop IBM and others from duplicating that process. So, on 8 April 1965, I filed for the sort patent. On 28 April 1968, a little more than three years later, I was awarded US patent #3,380,029 for "Sorting System." The awarding of this first

software patent warranted mention on the front page of *Computerworld* (see Figure 4), and *Fortune* magazine called it an "Unprecedented Patent."

After I filed for the patent, Mort Jacobs and I then turned our attention to the Autoflow product. Mort asked pointed questions to which I had never given much thought. How are you protecting the object code you are distributing to users? How are you protecting your source code internally? What rights are you giving your users in your contracts with them?

I hadn't been smart enough to ask the right questions—and I certainly didn't have the answers. But thanks to the long-standing business relationship that was developing between ADR and Mort Jacobs, he quickly provided the right answers. In 1965, Mort was responsible for ADR's copyrighting the code that it distributed and treating its source code as a trade secret. In addition, we attempted to patent any software technique that we believed reached the level of an invention.

At that time, IBM was distributing source code with its free software and putting both the object and source code in the public domain. To protect our IP rights in the program, Mort suggested we lease the program for three years at a time and call the program *equipment* because contract law was clear on the limited rights of a party when it leased equipment. Mort also drafted our patent for Autoflow, which was filed on 7 December 1965 and granted almost five years later on 6 October 1970 (see Figure 5, next page).

While marketing Autoflow for the IBM 1400s, we became aware of the IBM Flowcharter, an IBM field-developed program (FDP) that IBM distributed for free. This program was not automatic, and the flowchart it produced was based on separate, programmer-prepared input. Nor did it use the assembly language program as input. However, the IBM Flowcharter's availability was a major factor in many delayed or lost Autoflow sales. Our prospects went to IBM and asked for improvements to the free IBM program. It was widely believed that IBM would develop a similar type of program and provide the enhanced system to customers at no cost.

Mort Jacobs advised us on how best to communicate to IBM our concern that distributing

COMPUTERWORLD

THE NEWSWEEKLY FOR THE COMPUTER COMMUNITY

Vol. 2 No. 25

June 19, 1968

Price: \$9/year

First Patent Is Issued for Software, Full Implications Are Not Yet Known

WASHINGTON, D.C. — The first patent for a software computer program has been issued by the Patent Office to Martin Goetz, Applied Data Research vice president. This represents the first landmark after the withdrawal earlier this year of the proposed "Section 106" designed to outlaw patenting of computer programs.

The objections to the patenting of programs have been on two basic levels. Theoretically the main objective was that "programs were just mathematics and you cannot patent math." On a more practical level, it has been observed that there was no way of searching for prior art, as the computer community has not yet developed any effective way of classifying programs. This practical problem of searching has a secondary, but definitely real, impact due to the fact that the Patent Office is currently short of funds and presumably would not

welcome a flock of computer patent applications.

Inventing in a Garret

It appears possible that such a rush may develop as a result of the precedent set by the software patent. Before this, patents involved logic design of actual hardware which few programmers were able to do and which even lower could arrange to have built if the Patent Office wanted to see the system in operation.

Under the new procedure, it appears that a programmer working by himself, or in the traditional garret, may be able to get a patent and so exclude the hardware manufacturers from using his technique without his agreement. He still has to show novelty, and the traditional flash of genius, but the almost mandatory cooperation of one of the industry giants is no longer necessary.

Infringement Possibilities

In the meantime the current position is that any installation which is currently using the sorting technique proposed by Martin Goetz is legally infringing his rights—no matter where they got the program, whether from a manufacturer or from one of their own programmers.

The infringement appears to start when the program is loaded (because that effectively is a manufacturing process) and continues when the program is used. Whether the program is obtained by sale, is obtained free, or is paid for

7090 or User-Designed Codes Work on System

LOS ANGELES, Calif. — A new computer system was introduced here which, far from forbidding the user to change its microprogramming, encourages him to do so. The IC-6000E introduced by Standard Computer Corp. is particularly for universities and other educational institutions and comes with a number of different machine codes. These machine codes, which currently include those of the IBM 7090, 7044, and 1130 systems and a specialized Fortran system, are supported through a microprogramming system called Miniflow. The same Miniflow system allows other machine codes to be designed by the user to fit his own

Operating details of the IC-6000E appear on page 7.

sciences, linguistics, electrical engineering, etc., can create new computer systems at will and study their applicability to the particular type of problem they are considering. Changing the machine code from one version to another takes less than a minute and can be done without any engineering support. It is expected that particular researchers will create machine code operations that currently require macros and that as a result, they will get a higher efficiency for their own particular types of opera-

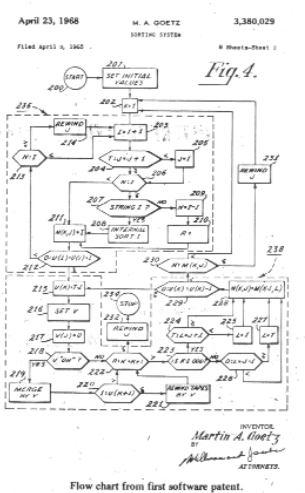


Figure 4. From its 19 June 1968 issue, *Computerworld's* front page announcement of the first software patent. (Reprinted with permission of *Computerworld* magazine.)

its free flowcharting program was anticompetitive. He wrote to Burke Marshall, IBM's chief counsel, arguing that they were misrepresenting their product when they called it "automatic." We argued that they were hurting our market by giving the software away for free. Mort put IBM on notice that they might be violating our patent application if they produced an automatic flowcharting program. But our efforts were to no avail; we got only lip service from IBM. These early skirmishes with IBM were the first in a long line of confrontations on the subjects of tie-in sales, monopoly, and patent protection for software.

With some success in the 1400 marketplace and the emergence of Cobol, ADR embarked on a major effort to develop Autoflow for the IBM S/360 assembly, Cobol, and Fortran languages.

We advertised extensively, expanded our sales force significantly, and were at all the major computer shows (see Figure 6, next page). From 1967 to 1970, ADR licensed more than 1,000 Autoflow systems. IBM reprogrammed its flowchart program for the IBM 360 and continued to be a major competitive factor with its free program.

By 1967, several companies were marketing software products, and the press was beginning

Oct. 6, 1970

M. A. GOETZ
AUTOMATIC SYSTEM FOR CONSTRUCTING
AND RECORDING DISPLAY CHARTS

3,533,086

Filed Dec. 24, 1968

38 Sheets-Sheet 1

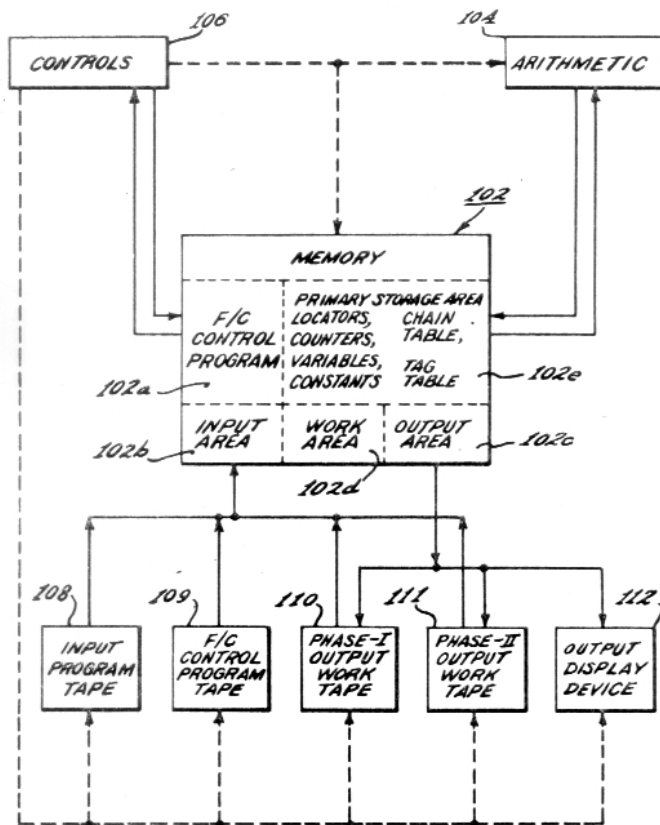


Figure 5. First page of patent that shows a schematic block diagram incorporating a control program for flowchart documentation. (Courtesy of the author.)

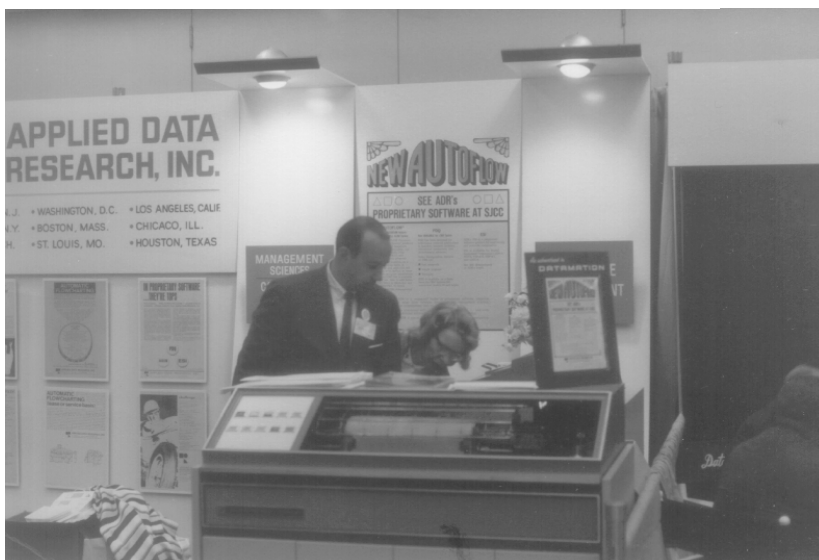


Figure 6. ADR and Martin Goetz at the 1968 SJCC exhibit hall promoting its newest version of Autoflow. (Photo courtesy of the author.)

to take notice, questioning whether IBM and other manufacturers would unbundle their software from their hardware. In the March 1967 issue of *EDP Industry and Market Report*, 17 software packages were profiled, including Autoflow. In that issue, editor Pat McGovern wrote about the potential of software packages, questioning if manufacturers would ever voluntarily unbundle, opening themselves to widespread competition.

With little expectation by anyone that IBM would voluntarily unbundle, there was a growing awareness in 1967 that the independent software companies had a golden opportunity. Because of ADR's difficulties in competing against IBM's free flowcharting program, I believed that ADR's best strategy to combat IBM was in building software that did not compete with IBM's free software. I abandoned any thoughts of building sort programs to compete against IBM's free sorts.

Industrial Research magazine recognized Autoflow as one of the 100 best products of 1967. Based on our continuing success with Autoflow and the growing acceptance of software products, ADR began investing in four additional products: the Librarian, ROSCOE (Remote OS Conversational Operating Environment), MetaCobol, and SAM (System Analysis Machine). Everything looked promising. ADR's stock rose, and the potentially huge marketplace for software kept expanding.

Meanwhile, having been burned by IBM's flowcharting program, ADR and other independent software companies complained to the US Justice Department that IBM was monopolizing the software industry.

I remember many meetings in 1967 and 1968 with Justice Department attorneys who were concerned about IBM's dominance of the mainframe hardware field. During 1968, the Justice Department interviewed a number of independent software companies and in January 1969 brought suit against IBM. The complaint, which covered IBM's dominance of hardware, also alleged that IBM had hindered the software products industry's growth as a result of bundling the hardware and software, beginning in the 1960s.

IBM had continued to announce and deliver free programs. During ADR's development of ROSCOE (an online program development system), IBM began the free delivery of CRBE, a Conversational Remote Batch Entry program, a direct competitor to

ROSCOE. Not long after, they announced CRJE (Conversational Remote Job Entry), an improved version of CRBE.

ADR's strategy was to avoid direct competition with IBM. We had focused our market niches in technology areas that IBM had previously avoided, but we had no way to know what IBM's future development plans were. We could not effectively plan for our future if IBM was allowed to continue bundling its programs with its hardware.

ADR sues IBM

In April 1969, ADR sued IBM for monopolizing the software products industry. We were attempting to protect our future and, at the same time, collect damages for reduced revenues in the flowchart market during the 1960s.

Just two short months later, in June 1969—six months after the filing of the Justice Department suit—IBM announced that, beginning in January 1970, it would unbundle its systems and application software except for system control programs (SCPs). These programs, said IBM, were an integral part of its hardware and would continue to be free. CRJE and all of IBM's operating systems fell into that exempt group.

In late 1969, shortly after its unbundling announcement, IBM added insult to injury by announcing TSO (time-sharing option), another ADR ROSCOE competitor, as an option for its free SCP MVS (multiple virtual storage) operating system. TSO was scheduled for delivery in 1971, 18 months later. ADR then faced a battle on two fronts: Not only would TSO be free, it would not be benchmarked for 18 months. There was great interest in ROSCOE, but few sales resulted. IBM contacted all our prospects and told them about CRJE and the forthcoming TSO.

The year 1969 was traumatic for ADR and for me. First, ADR brought suit against IBM in April. At about the same time, Dick Jones pushed to acquire Programatics, a West Coast programming services firm selling Pisort, a proprietary software program, in competition with IBM's free sort program. Against my objections and those of John Bennett, who had moved from Washington to Princeton to become director of marketing, the ADR Board approved the

THE PRINCETON PACKET

WEEKEND EDITION

VOL. 185, NO. 49

PRINCETON, NEW JERSEY, SATURDAY, NOVEMBER 15, 1969

FIFTEEN CENTS A COPY • \$3 A YEAR

Thursday's Miracle: All Safe In Crash, Fire



Building Hit By Plane Trying Emergency Landing

by JOSEPH DONADEU

No one was killed or seriously injured Thursday afternoon when a light plane crashed into the Applied Data Research building on Route 206 in Montgomery Township.

One spectator was heard calling out to a witness standing nearby, "Well, it's good your team was working today."

The small, single engine plane, piloted by Norman N. Kay, 56, of 1809 Pallas St., Philadelphia, was enroute from Sanford, Maine to North Philadelphia Airport when it developed a fuel leak in the engine tank, according to Transport Baron Landers, of the Princeton State Police Bureau.

Mr. Kay told police that his fuel gauge registered empty and he noticed fuel stains on the wing. He spotted the Princeton Airport from the air, and, according to police, attempted to land without radio contact.

The plane was making its final approach, into the wind, when he lost power and his engine cut out and he crashed into the southeast corner of the roof, bursting through to one of the upper floors in the building.

The point of impact was 400 feet west of the airport runway according to Transport Landers, and the building at that time was occupied by 150 people when the plane came through the roof.

The 2.1-ton, Cessna 441 in the office of Princeton Funding Inc., and the plane was dropped from the plane to the

office floor, State Police said. There were 10 employees at Princeton Funding at the office at the time of the crash, they added.

According to State Police and witnesses nearby, the plane's landing gear tipped over and black smoke was pouring out before the plane burst into flames. There was no explosion, they say, and the fire spread from the fuel tank.

Within seconds of the crash, employees of the three firms occupying the building were leaving the structure and moving their cars from the parking lot nearby. Many returned to retrieve coats, papers, or important materials.

None of the people in the building at the time were injured, according to police and hospital reports, and the pilot was reported to have walked into Princeton Hospital where he was treated for a cut above the right eye and a fractured arm.

Raymond Whitlock, Rocky Hill Fire Chief, an employee of Nantux Oil, was at nearby Princeton Motors, when a siren alerted him "my line, look" and pointed to a plane on the way down.

"I left immediately to get the spotter in heavy rain," Mr. Whitlock said, and within an hour I was back up the road with fire crew and fire equipment. We saw the first emergency in rain."

Among the other companies in (See PLANE CRASH, Page 15)



Figure 7. A 15 November 1969 newspaper headline about ADR's main office being hit by a plane. ADR's building was destroyed from the resulting fire. (Photo courtesy of the Princeton Packet.)

acquisition of Programatics. My opposition was based on wanting to stay out of IBM's path, not selling products in competition to them, if possible. And, because we had just introduced the Librarian, ROSCOE, MetaCobol, and SAM, our sales staff was already dealing with too many new products.

The acquisition proved both disruptive and costly. Dick Jones added Pisort to ADR's suit and tried to obtain a preliminary court injunction against IBM's distribution of its free sort. His effort failed, because ADR could not prove that the distribution would cause "irreparable injury," but the experience did afford us insight into what it would take to get a preliminary injunction against IBM in the future.

Then, in October 1969, ADR's main office building, hit by a plane, caught fire and was destroyed (see Figure 7). It was a miracle that no one was killed or injured. The next day we moved to temporary quarters and never lost a beat. One saving grace was that coverage of the fire gave our products some excellent publicity. We even developed an advertisement and brochure, "Unplanned Demonstration," describing how the Librarian had saved our source code and Autoflow had allowed us to reproduce our flowcharts. Shortly thereafter, *Business Week* published an article about ADR's unusual advertising campaign with the headline, "Turning Disaster into a Sales Pitch."

My growing involvement in IP and anti-competitive industry issues paralleled this 1968-1969 time frame in which ADR was

SELECTIONS OF 25,000 and buildings for in shared with as quarters at Thursday's Applied Data Research lines unobscure another small plane approach Princeton airport runways, just a few hundred feet away.

Fortress IBM?

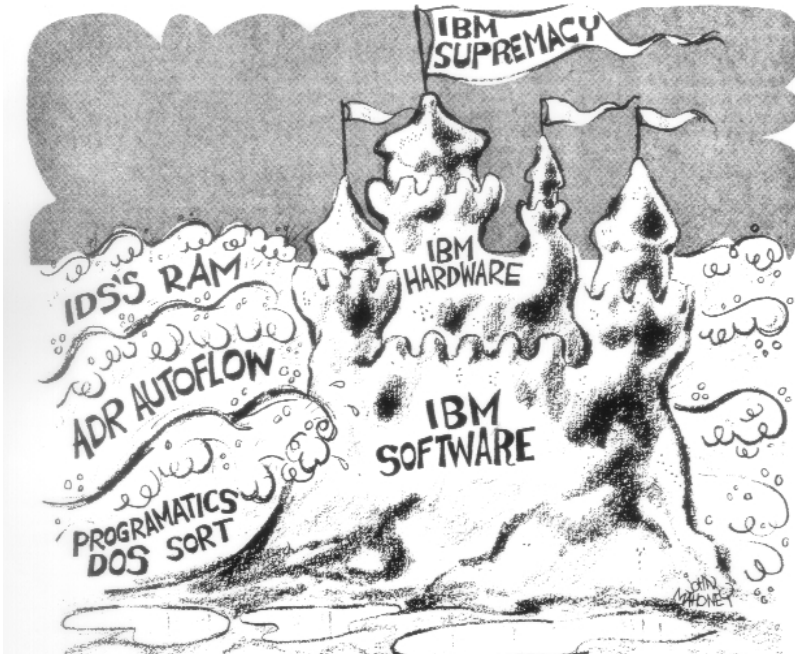


Figure 8. A 1969 *Computerworld* editorial cartoon, from its 14 May 1969 issue, depicting IBM's fortress as a castle of sand. (Reprinted with permission of *Computerworld* magazine.)

struggling to survive and become a viable company. In late 1968, I attended The Law of Software conference in Washington, D.C., where software's patentability was discussed in five papers, including one by the commissioner of patents, Edward Brenner, as well as one by Mort Jacobs. Three papers addressed the question of copyright protection for computer programs. Two papers focused on the business outlook for software and whether the bundling of hardware and software was an antitrust violation. These issues were urgent. The Justice Department suit, the ADR antitrust suit, and IBM's unbundling announcement all took place during the first six months of 1969. Mort Jacobs, Dick Jones, and I, totally involved with these issues, became more confident that IBM's supremacy of software was about to change (see Figure 8).

At a second Law of Software conference, in late 1969, I spoke on the future direction of software and the need for software protection. By that time, IBM had announced its software unbundling would take effect in January 1970. IBM presented a paper on its new software licenses in addition to a paper supporting the need to protect software. IBM, however, indi-

cated that it opposed the patenting of software, a position it had held since the early 1960s. The new commissioner of patents, William Schuyler, took a different position in his presentation—there would be new guidelines permitting software's patentability.

Software's taxation was addressed in a paper by Sheldon Cohen, who would later become the US tax commissioner. Software taxation is another area in which I became involved as the industry began to grow in the 1970s.

Following the conference, I can still remember the feeling of elation that ADR's (and my) views were beginning to garner support. The conference also renewed my confidence that our suit against IBM had merit and that IBM's distribution of CRJE and their announcement of TSO were unlawful. However, despite my great admiration for Mort Jacob and his law firm, I began to realize that his small firm did not have the expertise or manpower to effectively handle our suit or do what would be necessary to bring IBM to the negotiating table. So, in late 1969, we employed the

Malcolm Hoffman law firm to handle our suit against IBM. Malcolm Hoffman himself, well respected in antitrust circles, had written a book on antitrust law and had worked for the Antitrust Division of the Justice Department from 1944 to 1955.

Over the next several months, I worked closely with Hoffman and his staff to prepare our case against IBM's bundling of CRJE and TSO. I also worked with ADR's sales force to obtain evidence that CRJE and the TSO announcement were crippling ADR's ability to market ROSCOE.

Meanwhile, ADR's financial condition worsened. Not only were our costs increasing because of the lawsuit, but our cash flow remained negative in spite of paper profits. The Librarian and Autoflow sales continued to increase, but our other new products were not selling well. Most notably, ROSCOE sales in 1970 were near zero as IBM users tried to evaluate our product against CRJE and the forthcoming TSO. Per IBM's unbundling announcement, all existing pre-1970 software, including CRJE, would remain bundled (and free). TSO would also be free because it was to be part of the free IBM MVS operating system.

ADR's poor financial results in 1969 also reflected the fact that Dick Jones had opened 17 one-man mini-offices in 1968 and 1969 in anticipation of higher product sales. Although ADR's 1969 revenues of \$6.2 million were almost 50 percent higher than our 1968 revenues, our expenses had doubled.

Dick's last full year with ADR was 1969. The Board of Directors had lost confidence in his ability to control ADR's growth. While Dick had done an outstanding job in helping ADR grow in the 1960s, he did not have the discipline to work with budgets and to get Board approval on major expenditures.

In March 1970, having no desire to be president or to take responsibility for ADR's other divisions, I recommended to the ADR Board that John Bennett replace Dick Jones as president. The Board agreed. John accepted the job and Dick left ADR and moved to California. (Two years later, ADR divested itself of Programatics, by selling it to Dick.)

Shortly, in June 1970, ADR sought a temporary restraining order (and then a preliminary injunction) against IBM to stop its distribution of CRJE. Our sales force produced memoranda and affidavits from several users providing evidence that the free CRJE and the TSO announcement had crippled ADR's ability to consummate ROSCOE sales. And our poor financial condition, as it turned out, gave our counsel some needed and strong ammunition with which to argue our case.

A 60-day restraining order against IBM was granted in July 1970. A September 1970 date was set for an evidentiary hearing for a preliminary injunction against IBM's distribution of CRJE and the bundling of TSO with the MVS operating system.

But the hearing never happened. In August 1970, IBM approached ADR with a settlement offer and we settled our antitrust suit. ADR received a cash settlement of about \$2 million, and IBM agreed to market Autoflow in Japan and Brazil with a guaranteed minimum payment. IBM also agreed to license a large number of copies of Autoflow and the Librarian for its own internal use. This settlement provided ADR with some much-needed cash and provided for some foreign distribution of our products.

We continued to work closely with IBM after the settlement. We were a customer of its mainframe computers, we had a marketing agreement with two of its IBM World Trade subsidiaries, and IBM was now the largest user of Autoflow and the Librarian.

With John Bennett as ADR's new president, money in the bank from the IBM settlement,

the IBM's unbundling announcement in effect, a Justice Department suit against IBM, and ADR's base of more than 1,000 existing customers and many new products, ADR had a new lease on life. Our stock, which had hit a high of 40 in 1968, was down to five, but we were still in there, fighting for survival.

John Bennett gave me profit-and-loss responsibility for the Software Products Division of ADR and I had my work cut out for me—what had been John's product sales force and marketing responsibilities were now mine. For the first time, I had control of ADR's software products business.

In 1970, the software industry emerged with lots of opportunities. But would IBM and the other computer manufacturers dominate this industry and smother ADR and those hundreds of small independent software companies that had just formed? As it turned out, there was strength in numbers—several early software companies recognized the power of trade associations.

ADR had been active in a small software trade association called the Association of Independent Software Companies. AISC was formed in May 1968 by 11 software companies (see Figure 9, next page). Some of the companies were in the software products and programming services business; some, exclusively in services. Although a small association, AISC was active in developing white papers on software protection through patenting, on fair competition issues from not-for-profit companies like System Development Corporation, and on unbundling of software and hardware by IBM.

Dick Jones had been ADR's representative to AISC, a role I filled after his departure in early 1970. Although many new software products and services companies had formed after IBM's unbundling announcement, AISC found it difficult to attract new members. Meanwhile, the Association of Data Processing Service Organizations (ADAPSO), another trade association made up of hundreds of small service bureau centers, was hoping to expand by forming new ADAPSO sections and asked Larry Welke, president of International Computer Programs, to form a new software section. ICP, which produced and marketed a catalog of software programs, was considering joining AISC but decided instead to organize the ADAPSO section.

The Software Section of ADAPSO was formed with 26 software companies in October 1971, with Larry Welke as its first president. I had known Larry for several years, and we soon opened discussions on the possibility of AISC also joining ADAPSO. In the spring of 1972,

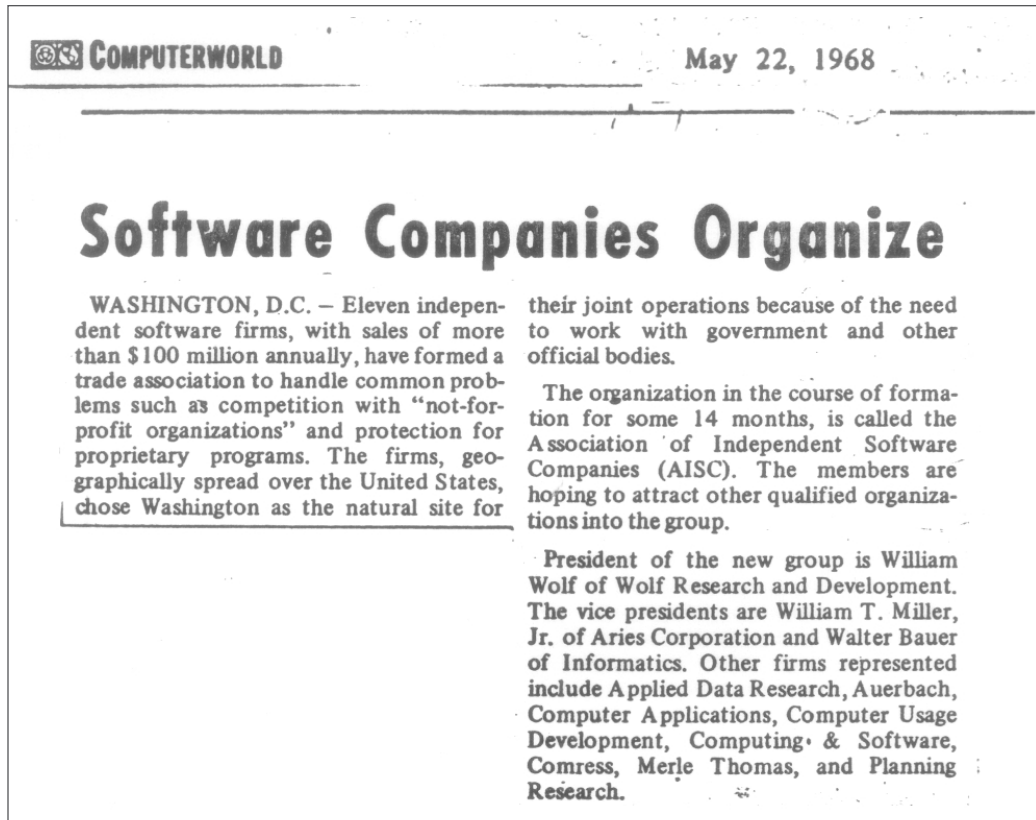
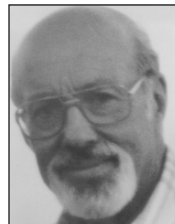


Figure 9. A *Computerworld* article, from its 22 May 1968 issue, announcing the formation of the AISC trade association. (Reprinted with permission of *Computerworld* magazine.)

AISC agreed to merge with the software products section of ADAPSO and name the expanded section ADAPSO/AISC. The five executive board members of AISC (of which I was one) agreed to be on the new section's Board of Directors, which asked me to form an ADAPSO committee on software protection. (In 1973, the software section of ADAPSO was renamed the Software Industry Association.)

(End of Part 1. Part 2 will feature Goetz's role in growing ADR to a \$200,000,000 company and in ADAPSO, which dealt with industry issues as IBM began to exert its enormous marketing powers and resources in the licensing of its program products.)



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.

Readers may contact Martin Goetz at mgoetz@nis.net.

For further information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.