ADAPSO Reunion Workshop: Intellectual Property

Moderators:
Ronald Palenski and Bill Aspray

Recorded: May 4, 2002
Washington, DC

Table of Contents

ADAPSO Reunion – Intellectual Property Workshop

Conducted by Software History Center—Oral History Project

Abstract: A group of former computer software and services company CEOs and attorneys met to discuss the issues involved over the years in protecting software. They talked about patents and copyrights and the difficulties in using them to legally protect software. This was not a major issue in the early years when most application software was custom, but it became significant with the use of packaged software and the introduction of widely-distributed software for the PC. They discussed the difference between published and unpublished copyrights and how IBM became more concerned with the issue after they unbundled their software. They finished by discussing the characteristics of the early period of software and what they expect will happen to the industry in the future.

Participants:

Name	**Affiliation**
Bill Aspray	Computing Research Association, Historian
Martin Goetz	Applied Data Research (ADR)
Thomas Haigh	Historian
Lee Keet	Dun & Bradstreet, National CSS
Joan Krammer	Smithsonian
Ronald Palenski	Former ADAPSO General Counsel
Oscar Schachter	Advanced Computer Techniques Corp.

Introductions

Ron Palenski: I'm a former General Counsel of ADAPSO, and I'm going to moderate the session.

Marty Goetz: I'm formerly president of Applied Data Research. I'm a private investor.

Norma Goetz: I'm Marty Goetz's wife. Been there, been through all of this.

Oscar Schachter: I was an active member of ADAPSO with Advanced Computer Techniques Corporation for about twenty years and am now in private practice as an attorney doing mostly intellectual property licensing work with software companies.

Lee Keet: At the time that we're discussing, I was President of the software products group at Dun & Bradstreet, formerly of National CSS before its acquisition in 1979 by Dun & Bradstreet. Since 1984, I've been President of Vanguard Atlantic Limited, a venture fund.

Nancy Keet: I've been on the sidelines watching all of this unfold.

Joan Krammer: My involvement is through the Smithsonian and I'm here to make summary notes.

Bill Aspray: I'm the executive director of Computing Research Association and, I guess, the Vice-Moderator, in case there's any vice.

Palenski: Marty, you were Chairman of the Software Protection Committee during what period?

Goetz: From 1972 to 1980, I was Chairman of the Software Protection Committee of ADAPSO. And according to the research I've recently done for my memoir, Lee Keet became Chairman in 1980.

Palenski: If you don't mind, I'll ask you some of the questions that have occurred to me throughout the years. I put this timeline together, if for no other reason than, perhaps, to jog memories.

Areas of Intellectual Property Law

To me, there seems to have been two major areas of intellectual property law that were relevant, at least from a federal statutory viewpoint, back in the 1960s and the 1970s: patents and copyrights. There seems to have been, for example, a lot of churning in the patent area back in about 1965-1966. There was a commission established by then-President Johnson to take a look at the question of patentability of computer programs and, at least as I understand it, the commission recommended against it, primarily because of the administrative burdens in collecting evidence of prior art.

In addition, there was the phenomenon that the patentability of computer software was disfavored by the U.S. Patent and Trademark Office, but more favorably received by the Court of Customs and Patents Appeals. Thus, software-related patent applications were frequently rejected by the Patent Office but would be appealed to the Court of Customs and Patent

Appeals, which tended to be more sympathetic towards patent protection.   Then some of these cases went up to the Supreme Court and, until Diamond v. Diehr in 1981 or thereabouts, the Supreme Court had never upheld any of these. So, a lot of churn in the patent area.

In the copyright area, the Copyright Office would register programs under the Rule of Doubt beginning about 1964, but it wasn't until 1978 that you had a clear statement in the Copyright Act that computer programs were copyrightable.

Marty, because of your familiarity with the patent issues, I'm wondering if you might talk a little bit about which early ADAPSO companies were involved in the software protection effort? What were some of the discussions you had, decisions you made, and things that you did?

ADAPSO's Interest in Software Protection

Goetz: When I got involved in ADAPSO with software protection, my interest was in how you protect your software assets. At that time there was a suit against IBM and we were interested in lots of things besides intellectual property per se. And that's why it was called the Software Protection Committee as opposed to the Intellectual Property Committee.

I got involved in the mid 1960s when, by chance, I ended up getting the first software patent. There were two very famous cases, one that evolved in the 1960s called the Prater versus Wei case, which is well written up. And a second case in early 1972 which was called the Benson case. Both of those cases preceded Dann versus Johnson. Since about 1965, when I got the first patent, there had been a controversy as to whether software contained patentable subject matter. There was a lot of confusion because people thought of it as getting a patent for a program, which was not the case, because my patent, for instance, was for a sorting process.

So there was always confusion whether you should copyright or patent your programs and they really were separate issues. For at least fifteen years, from 1965 through 1980, the patentability of software was in question. There was also trade secret as another possible method of protection.

I got involved with the Software Protection Committee after the presidential commission that you alluded to. IBM was pushing their position which, at that time, was that they didn't believe that software should be patentable. They weren't selling programs, they were putting programs in the public domain. IBM, which for many years got the most patents of any company in the United States, or in the world, was against the patenting of software.

There were some initial guidelines back in 1965 on how to patent software but it kept changing depending on the patent commissioner, and then on the cases that came up. So for fifteen years there was lots of confusion.

Meanwhile, unbundling happened and lots of companies were being formed. I got involved in the software protection area but patenting was only one of my interests. My primary interest was how to effectively compete against IBM.

Palenski: What position was the association advocating at that time with respect to patents?

Goetz: Well, before I became part of ADAPSO, I was involved with an association called AISC which had, I believe, supported the Prater v. Wei case.   I brought that effort with me when I joined ADAPSO, which at the time was in the process of being an amicus in the Benson case.

ADAPSO was generally supportive of the patentability of software but it wasn't universal. There were some companies within ADAPSO, IBM and others, that thought patenting was not appropriate and supported other forms of protection. In particular, Lee Keet was very much for trade secret protection as opposed to patent protection.

Schachter: Did IBM eventually change their position on the patentability of software? Do they now apply for software patents?

Goetz: IBM was filing for patents continuously, at least when the window to do so was open—it closed for awhile in the 1970s—but they were consistent in being opposed to the patentability of software. In fact, in the late 1960s or very early 1970s, they came up with a registration system as an alternative to patenting, somewhere between a patent and a copyright. But, while they were against it, they were still filing lots of patents and continue to do so.

Aspray: You talked about things that were happening in the courts and about the Presidential Commission in the Executive Branch. Was there anything going on in the Legislative Branch that was trying to shape what was happening?

Goetz: Yes, there were some committees that were formed. I don't remember them.

Aspray: Any major pieces of legislation?

Goetz: There was no legislation until the late 1970s.

Palenski: In 1976, Congress adopted a new Copyright Act that became effective in 1978. Before that, the 1909 Copyright Act was in effect. From what I can tell from looking at Marty's records and from what I recall, the association did file briefs in most of the major patent cases that came before the Supreme Court: Gottschalk v. Benson, Dann v. Johnson, and then Parker v. Flook.

The ADAPSO position, as I recall, was that a process which is otherwise patentable if implemented in hardware ought to be equally patentable if implemented in software. So the association really didn't talk in terms of "software patents" but used the more traditional patent terminology.

Goetz: In the late 1960s and early 1970s, software companies started to file for patents. Boole & Babbage, a software products company, filed an early software patent. Today there are hundreds of software companies filing for patents. Perhaps too many. Business method patents are still considered very controversial. A business method is quite different from a really unique process and, unfortunately, all of the cases that came before the Supreme Court were not representative. They really were marginal, like a computerized system to set off an alarm. They didn't represent the type of things that software companies were trying to protect.

<u>Companies Involved in Software Protection Efforts</u>

Palenski: What were some of the companies involved in the software protection efforts back in the late 1960s and early 1970s?

Goetz: I think Oscar Schachter's company, ACT, was.

Schachter: It was one of the reasons we joined ADAPSO in the early 1970s. Software protection was one of the key issues that motivated us to join. We were primarily interested in traditional trade secret protection, probably as much from a lack of funds as anything else. Secondarily, we were interested in the copyright protectability of software. We weren't really focused on patentability of software.

Goetz: One of the problems was that software was always sort of looked on as a second-class citizen. It started out being a free service put in the public domain. And, really, part of what we were trying to do was to get stature. Every other industry seemed to have patent protection but here was an industry where you couldn't get patent protection. We were fighting IBM competitively at the time. So if IBM was against something, we were for it automatically. [*Laughter*]

That didn't turn out to be the case in the copyright area but it seemed like, wherever we were going, IBM was on the other side.

Aspray: Were there particular kinds of software companies for which this was a more important issue?

Keet: There were no application software companies, as I recall, on the committee either when you were running it, Marty, or when I was running it. Most of them were system software companies.

I was a member of the committee for a number of years before I took over the chairmanship and, as I recall, the members were all systems software companies.

Aspray: Did the application companies have some different mode of doing business or some other kind of protection mechanism so that they didn't have to use patents and copyrights?

Keet: A lot of the debate in the early years grew out of a lack of understanding or knowledge of where the courts and the legal system would go. There was a lot of discussion about needing additional protections because we weren't sure we were going to really get the protection that many of us thought were embedded in copyright or, potentially, in patents. There was a large debate about how so much software is not obvious, doesn't implement an algorithm, etc. My position all along was that there was going to be a very narrow window of opportunity for somebody to find something that was truly patentable software.

But our discussions really revolved around how to protect the distribution of our software, how to protect it against theft by customers, protect it against theft by rogue employees, and theft by our distributors. The application software companies in those days were much more service companies than they were packaged-goods companies and, therefore, they were working more closely with their customers. In retrospect, the protection we relied on the most was the business-to-business contract. It turned out to be the ultimate protection until people started to package their software for easier distribution, which coincided with the PC revolution.

At the end of my tenure as chairman of the committee, we had Jon Shirley, the President of Microsoft, involved. The PC guys couldn't get the same contract protections. They weren't delivering hands-on services so they weren't there to observe if somebody was using their product illegally. Their interest, therefore, was much more in how to protect a mass-produced piece of goods.

So there was a transition to needing to know what the legal protections would be, instead of focusing on alternative ways of protecting ourselves. Are we going to enjoy copyright protection for what we've done? Are we totally reliant on trade secrets and contracts? Is Marty going to prevail and everybody's going to be able to get a patent? It was confusion.

Goetz: Definite confusion and, of course, to the extent that one could get a patent, a very small number of programs would meet the criteria. It was very unclear how you would stop someone from using your program. It was unclear whether copyright law was actually valid. It was never tested in the courts. So, back in the early 1970s, the focus was on making sure your contract was valid. In fact, that was one of the reasons companies were licensing rather than selling the software because you'd lose lots of rights if you sold it.

Schachter: Just to add a footnote to what kinds of companies were patenting software. The systems software companies were a lot closer to the machines, so to speak. They were actually driving the machine. Theirs was a process that was making the machine work in a particular way and I think that they could more easily formulate the words that would allow them to patent a particular process. The application companies were one step removed from the machine and from the system software and, therefore, had difficulty in formulating the kind of terminology that allowed them to patent software—until you get into the business process patents that we have today. So that may have been another reason why the early companies were all pretty much system software companies.

<u>**Difficulties in Protecting Software**</u>

Keet: I'd like to add that we didn't even know ourselves what category to put software into. There was a lot of discussion about whether this was analogous to the piano roll. And, therefore, do we fall into that category? Or is this analogous to a machine component or part, and do we fall into that category? In other words, we were trying to find the bucket that we belonged in because we were inventing stuff and we didn't really have any history to look back to. And we weren't getting tremendously good advice from the attorneys because some of them had a particular penchant for patents and others thought that we really needed to rely on trade secret protection. And there were some who were saying that it's unique and we need sui generis legislation. I'm trying to remember the name of an attorney... Was it Pamela Samuelson?

Palenski: Well, there is an attorney by that name.   Formerly on the faculty of the University of Pittsburgh Law School, she's now teaching at Stanford University Law School.

Keet: She was leading the charge in the early 1980s. She was leading the charge for: Forget it, guys, copyright doesn't apply, patent doesn't apply, we need a legislative act to protect software. I remember some very heated meetings where we had people yelling patents, we had people yelling copyright, we had people yelling trade secrets, we had people yelling contracts, we had people yelling sui generis. And I'm sitting there thinking, why did I take this job? [*Laughter*]

Palenski: Part of the issue was that, under the 1909 Act, which wasn't superseded until 1978, there was a Supreme Court case, White Smith versus Apollo. It ruled that pianola rolls, another digital medium of recordation, were not copyrightable because they did not communicate with a human being but, rather, a machine and, hence, were not "writings" in the Constitutional sense. And even after Congress passed the new Copyright Act under

Congressman Kastenmeier's leadership in 1976, it wasn't until Apple v. Franklin was decided by the Third Circuit that that argument was laid to rest, even though that case never went up to the Supreme Court. I can still remember the amicus briefs asking how something could be copyrightable if it wasn't designed to talk to a human, if it was designed to talk to a machine. So even though, beginning in about 1964, the U.S. Copyright Office would register programs, it was under the Rule of Doubt. Which means, essentially, we'll register this stuff for you, but, in the event of a litigation, you have to prove it's copyrightable.

Changes to the Copyright Law

The process to change that began in 1976. The CONTU Commission was set up to study the issue of copyright protection for computer programs. The computer program amendments to the Copyright Act, which implemented the CONTU recommendations, weren't enacted into law until December 1980. So you had some sixteen years where, with respect to copyright, things were very much up in the air.

Keet: Even after the implementation of the act, it was in doubt. Which is why we put a proposal before Kastenmeier in January of 1983. Software was not an enumerated art in the act that resulted from the CONTU commission. We felt that we needed it to be enumerated. I believe that the proposal for the improved protection of software resulted in a bill that was put before Congress, which was—help me, Ron—HR 6983?

Palenski: Yes.

Keet: It was simply a proposal to make a very, very modest change to the 1976 Copyright Act, to enumerate software as one of the copyrightable forms of expression.

Schachter: Originally, I believe, the Copyright Office would not accept object code as copyrightable. You had to provide source code and you had to provide the full scope of the program—which meant it would be publicly accessible. So no one was particularly interested in using that method of protection. It wasn't until they adopted later rulings which allowed you to file just pieces of the program, and to excise any real trade secrets even in those pieces, that people started feeling comfortable about registering software for copyright protection.

Goetz: I got indoctrinated very early by Mort Jacobs, my patent attorney, who had previously worked at the Patent Office and then had worked for RCA and then was in private practice. He kept saying, "Use the existing laws; don't wait for Congress to make something happen." Certainly, as I look back, to get Congress to enact a new law when you've got opposing sides is almost impossible. The CONTU committee got them to change two sentences in the copyright law to say explicitly that software is copyrightable material. That was sort of the extent of it. It was important for them to say that, to get that into law, but to get a new law would have been very difficult. And then you've got the people on the patent side who say, "Look, the patent system has been great for the last two hundred years and doesn't need to be changed." Of course, in the early 1900's, some famous person said there's nothing more to patent. Do you know who that was?

Keet: It was the then head of the Patent Office who said that everything that could be patented had been patented already.

Anyway, this proposal was to broaden the definition of what a computer program was so that it would include the schematics, the logic flows, and so on. That was the definition that was originally put forward in the World Intellectual Property Organisation proposals. But when they enacted the 1976 copyright law and the subsequent modification in 1980, we felt they had it too narrowly defined and, therefore, the courts could interpret the forms of expression that created software, such as source codes or block diagrams and so on, as not protected.

But then there were the court cases that came along and the courts, I have learned over the decades, get it right eventually. And that's what happened, they got it right.

Palenski: "Eventually" being the operative word.

Keet: Yeah, but the point is they respond to the needs of our society in a continuous fashion whereas Congress responds in a static fashion. Our industry is very dynamic and, therefore, a lot of those court cases actually went beyond what we could have done in a static bill because they had to make the decision to divide the baby or whatever. They had to say, "Is this right or is this wrong?" I was an expert witness in a number of cases and they always hinged on some piece of minutia that required a judge or a jury to think through what was the intent, and whether the law applies rather than what you would get by statute. So I've become a bigot on the other side of the issue, saying that we probably should have been using the existing laws more aggressively at the time.

Palenski: One impression I have, and maybe you can either affirm or deny it, was that IBM was a major proponent of copyright particularly because of the existence of the international conventions. So if you brought computer software under the ambit of copyright, then you would have automatically achieved international protection because of the Berne Convention and the Universal Copyright Convention.

<u>**PC Software Companies and Copyright Protection**</u>

Keet: I can't emphasize enough that when Microsoft and other PC companies joined ADAPSO that was their view also. They wanted enforcement of copyright. They felt that they could deal with the piracy issue as copyright infringement.   Piracy means a lot of different things to different people. I never had a customer steal software from me. We always relied on trade secrets and contracts, and there were always top-level companies behind them. I think that was probably true for most of us in the business. I did have a distributor steal from me once by making illegal copies of our software and selling them without telling us. It was the only case I ever had where somebody actually did that.

But with something like Microsoft Windows, there were whole shops being set up to stamp out copies and sell them by the tens of thousands at low prices. That's where those guys wanted to go. They wanted to use the power of international law and put some muscle behind it so they could go to the government of Singapore and say, "You've got a factory here that's stamping out illegal copies and you're in violation of world conventions." They didn't want to say, "You're in violation of US law."

Goetz: The only reason that IBM was slow to support copyright was that they were originally calling all their programs a service that they were giving away and putting in the public domain. So it didn't really make sense for them to say it's copyrightable. As soon as IBM unbundled, they started copyrighting their programs that were not in the public domain and I think just about every company used the copyright system. Perhaps a few didn't, but it didn't stop them from using trade secrets which were state laws as well.

Of course, then you didn't have consumer software and all the thievery. As Lee said, there was very, very little unauthorized use of programs by corporations. Now, of course, there is actually a lot of unauthorized use of PC software. So the Microsofts of this world do go after corporations.

Protecting Software with Software Locks

There were other ways of trying to protect software: by having software locks, a special device that you would put into your computer; having software keys that would be triggered with an internal identification. There were all kinds of techniques to stop illegal copying but many of them proved to be impractical.

Schachter: It created a lot of controversy in the industry when software companies tried to use those software locks and software keys to prevent copying.

Aspray: Why was this controversial?

Schachter: Well, the users were up in arms about it. If they wanted to move a piece of software from an obsoleted PC to another PC, they couldn't because it was registered for a particular PC. So the large corporate end users were very much against this.

Published vs. Unpublished Copyrights

I'd like to raise another topic, the question of published versus unpublished copyrights. A lot of companies relied on the protection as an unpublished copyright rather than going to the copyright office to actually register their copyright.

Palenski: Until the 1976 Act, there was a dual copyright system. In order to get federal copyright protection, a work had to be published with notice of copyright—the “c” in a circle symbol. The effectiveness or the applicability of copyright with respect to programs was in doubt and, therefore, companies relied on trade secrets. If you had to publish to get copyright protection, would you be foregoing your trade secret protection? And then there was also a separate state copyright system, also called “common law copyright”. One thing the 1976 Copyright Act did was to abolish the state copyright system so that you have a unified system now. The current Copyright Act protects both published and unpublished works, but that wasn’t the law of the land until 1976.

Schachter: And that became very important because one of the problems that I had with copyright registration is that the day you register a piece of software for copyright, your client is changing it. And a month later, the likelihood is that some of the code has been changed. Do you keep on filing updates of the software? How frequently do you have to file those updates?

So the use of trade secrets and unpublished copyrights is probably more prevalent in the industry. I don’t know if anyone has done a survey of how many software companies actually file for registration with the copyright office for every piece of software that they write and how many of them don’t rely on actual registration.

Goetz: Well, ADR, for example, never filed. We would always have a copyright stamp on our manuals and on our source code, on just about everything, but we never registered it with the Copyright Office.

Schachter: Yeah, ACT never did. I recently had an interesting case of a bank loan being made to a software company. They wanted to have a lien on all of the assets of the software company—trade secrets, copyrights, etc. And they were alarmed that the principal software of the company had not been registered in the Copyright Office. Although the company had an unpublished copyright and trade secret rights in the asset and were willing to give them a lien on that, the bank insisted that we file all of those pieces of software in the Copyright Office so that they would have a lien on a registered copyright. They said the bank just doesn’t like to have liens on unregistered, unpublished works.

Keet: It’s also important to point out that before the 1976 Act, there was a standard that said that you had to put your copyright mark on every printable page of a copyrighted work. We went to great lengths to generate a copyright notice in the delivered code to the customers because of the copyright mark requirement. Our attorney advised us that we had to show the copyright mark not just on the manual but on the actual delivered code and it should be on every page of the printout.

Fortunately, we had a source code generator and we modified it to put the copyright notice all the way through all of the source code. It was a mess. You would be reading along in the source code and there would be a copyright notice as a comment. It was just a bizarre time. The word I would use is paranoia. We, as an industry, I think, were paranoid about things that didn't happen. We thought that there was going to be a lot of thievery. We wanted to anticipate it because we put so much of our money and intellectual energy and effort into building these things, and we didn't want them stolen.

It wasn't until the PC industry came along that it actually turned into a huge problem. I can't emphasize that enough. I don't think you can find somebody from the era of the 1960s or 1970s that will tell you that they had a big problem with thievery.

Goetz: We didn't have any problem with the copyright because we didn't distribute our source code. So it was pretty clean for us. We delivered a tape to the user which had our object code, and on the tape and on the label it said copyright with a date. Our manuals had the copyright notice on the inside cover and that was about it. If anything got printed out on a printer, we didn't worry about it. So it was nice and clean. We didn't know if it really protected us but it didn't become a burden within the company at all.

IBM and Copyrighting

Keet: IBM never put a copyright mark on its software. I published a lot of software while I was at IBM, including some products that became tremendously widely used like the Bill of Material Processor. We put a copyright mark on the manuals. These were Type II and Type III programs and the documents would say "Copyright IBM Corporation" with the date. But the programs themselves never had a copyright notice.

Goetz: But that was in the 1960s.

Keet: Yes, this was in the early 1960s.

Goetz: They weren't copyrighting their programs because all of their programs were in the public domain, or most of them were.

Keet: I believe that the manuals said—we'd have to go back and find some of these in the Charles Babbage Institute archives—that the programs were the property of IBM and were provided to the customer for their exclusive use. But they never said these are copyrighted nor was there a copyright mark on the program.

Palenski: Under White-Smith v. Apollo, the Pianola roll case, the copyrightability of programs in the 1960s would have been up in the air—because the Supreme Court said if it doesn't talk to a human, it can't be copyrightable.

Keet: I'm just trying to paint the picture of a lot of confusion, a lot of paranoia, because we didn't know whether we were marching out there to sell our stuff to a bunch of thieves or good guys. It turns out they were all good guys.

Goetz: It was a very unclear era. There were the questions of whether software was tangible or intangible and what was software. Of course, IBM was giving it all away for free, and then suddenly they're selling it. What were they selling and how do you protect it? There was the question of: Is software taxable, is it tangible? There was a great deal of confusion all wrapped up with the intellectual property issues.

Keet: On the Contracts Committee, in which I participated and so did you, as I recall, Marty, we were trying to determine what should be the standard form of contract, or what is a good legal form for delivery of this product to a customer, and how do you protect it with trade secrets and so on in the contract. That took place simultaneously with these intellectual property issues.

Software and the Uniform Commercial Code

Palenski: In contract law, one of the big questions at the time was whether software was a "good" under Article 2 of the Uniform Commercial Code, which is the commercial law that has been adopted in all States except Louisiana. Or whether it was some sort of "service" or something else and fell outside of the UCC. For tax purposes, the question was whether software was tangible and, therefore, its transfer subject to state and local property taxation or sales and use taxation. Interestingly, the industry was whipsawed by government on the nature and taxability of software. On one hand, the Federal government took the position that software was intangible and, therefore, did not qualify for things like accelerated depreciation, the investment tax credit, and other favorable federal tax treatment. But the states, in their desire for revenue, were taking the position that software was tangible and, therefore, its transfer or sale was subject to sales and use taxation.

Keet: The UCC gave implied warranties for fitness and use for a particular purpose, as I recall the words, but we in the industry were trying to separate the transfer of the license for the software from our service agreements. The reason for that was, speaking for my company, we were trying to create hell-and-high-water leases for the underlying product. A hell-and-high-water lease is a lease where the customer agrees not to assert any defenses that he might have or to offset any payments that might be due against other claims. We actually had two documents. One was a service agreement and a warranty that was document B. And we had the software term license or lease, which was document A. The reason we separated them was we could take that pile of document A's and go to a bank and get financing for them because we had General Motors saying I will pay x dollars per month for the next thirty-six months no matter what. And the other document where we said we would provide service to General Motors was our liability and General Motors had to think that we were strong enough to stand behind that without being able to go against the financing lease.

Schachter: Was that the first case of off-balance sheet liabilities preceding Enron?

Keet: Oh, gee, did I start Enron? [*Laughter*]

Well, we would not have survived without those financing leases. I was very involved in these ADAPSO committees because of the very things we're talking about. If the UCC provisions applied under state law then we wouldn't have been able to separate the warranty. The position we took was, if it's a business-to-business transaction among sophisticated parties, that's not subject to the UCC laws. But the states individually did not necessarily agree with that position and therefore our financing was in jeopardy depending on the state in which we were writing those contracts. I just raise that because this was all tied together. You can't separate contracts from taxability, from financial issues, and from software protection. We were just struggling to figure out the business model that would allow us to keep our companies alive because there were only two sources of capital. One of them was from entrepreneur investors meaning friends, family, relatives, and anybody else you could steal a wallet from. The other was your customers, which meant you had to get sales and collect the money. We were really working hard to try to figure out how to make those collections come faster, make the amount of money we were getting bigger, and to finance it once we got a commitment.

Krammer: What I'm curious about is the comment that there was a problem anticipated that didn't materialize during that time period.

Goetz: We were selling to major corporations and then, maybe, smaller corporations but zero software to consumers. There was a contract and these were corporations where you're dealing with the data processing manager. He's not going to cheat because he could just end up losing his job. So there was very little thievery. On the other hand, you wanted to protect the program from your competitors. You didn't want your competitors to take it, change it and then go out and resell it. But from the user perspective there was very little thievery. In fact, ADR was never aware of any company that was using our software without being authorized to use it.

Keet: There was a problem which we discussed a lot in these committees of what you do about employees' non-compete agreements which were gradually being obviated by the courts, especially on the West Coast. California eventually made them totally useless. There were problems and lawsuits when an employee would go away and start another company to produce a competitive product and the competitive product looked awfully similar to the product that his previous employer had been producing. We had a lot of discussions about what we should be doing about that, what kinds of agreements could we get with the employees that would let us protect our property, and if we could rely on copyright, trade secrets, etc. for that kind of protection. So that's where the problems did occur.

Krammer: But I'm also curious to hear what other surprises happened. Going back to your comment that you were anticipating that something would happen and it didn't. What other surprises were there in that time period?

Keet: Well, we had a distributor in Germany which was a subsidiary of a larger German company based in Munich and the guy who was running it turned out to be an out-and-out crook. We distributed source code, unlike ADR, and our source code was customized to each customer. One of the things that we did was put the customer name into the source code so that when they actually compiled it and started the system it would say "XYZ Steel Company" on it. And we got a complaint back at our headquarters, bypassing the distributor, saying our distributor can't seem to get the right name on our software. [*Laughter*]

That's when we found out that there were lots more copies of Taskmaster in Germany than we actually knew about. That was the most egregious case of theft and, in that case, nothing would have protected us because it was like going out and finding your car. You might have every lock in the world on that car but if it's gone, it's gone.

Employee Agreements

Aspray: Coming back to Lee's comment about employees. Did ADAPSO or other trade organizations put out white papers or guidelines that suggested to companies what their model agreement should be or their model practice should be in these areas?

Goetz: We had employee agreements and they would generally protect us to the extent possible. The question was how they would hold up in your state which was always an issue. I don't recall ADAPSO getting too involved with employee agreements.

Schachter: I don't think we had a model employee contract. I think most software companies, after a period of time, adopted what is typically called now a professional practices agreement that they have virtually every employee in the company sign. It deals largely with confidentiality of materials, confidentiality of customer material, and the ownership of the work that you do. It clearly covers all inventions etc. etc. that you develop while you are working for the company. Some companies also have a non-compete agreement, but, as Lee said before, those are very much subject to state law.

I'm interested in returning to an earlier topic which relates to selling your software to large companies and, therefore, protection wasn't an issue, because you had contract protection. In the mid-1990s, there were a number of cases where companies were, in fact, allowing the distribution of illegal copies of PC software throughout the company, and the SPA went after them on that.

Today we have the situation where much business software for mainframes or for large servers is priced on the basis of the number of users on the system. You don't have to make copies of the software. You just put someone else on the system and—voilà—they're a user and they have a copy of the software available to them without your having copied it.   Software companies are struggling to protect themselves against the situation where you license the software for fifty users and suddenly there are a hundred users on the system. You can put locks into the software but that gets to be very impractical. The end user says, "I don't want to have to come back to you, and revise my contract every time I want to add five more users." Since most companies have access to their customers' computers for maintenance purposes, they're developing methods of actually policing the number of users which a large customer has using its software.

Keet: To the earlier question, one of the best things that ADAPSO did was create open forums where people would sit around and do exactly what we're doing here and trade experiences and ideas. On the employee issue, we had a lot of discussion about what others were doing and, from that, I learned the idea of putting in your employee handbook or in some other document what the rules and regulations are for the company. We got the employee to acknowledge that and made it into a contract by having the employee sign a postcard-size form on hiring. And also any time the employee got a promotion or a raise or a change of status, because there has to be consideration on both sides to make it into an agreement. We specified that violation of trade secrets or theft or misuse of copyrighted materials would constitute egregious damage to the company, etc.   Did ADAPSO publish anything on that? No, but it was pretty well accepted practice. I always used it in our company. Subsequently, since I started Vanguard, we've used it in every single company we funded right to this day. It has become kind of an evolved industry standard, but it came out of those discussions at ADAPSO, I want to emphasize, even though ADAPSO didn't publish a contract.

One of the things we're not focusing on at this conference as much as we should, in my opinion, was how the business practices evolved. We've spent a lot of time talking about ADAPSO and trade associations, which is appropriate. But if you go back to those early discussions, a lot of them had nothing to do with the formal committees or sessions. It was sitting around saying, "How much are you paying your sales guys? What's the compensation plan? How much do you expect a top guy to make? What are you doing in the way of marketing? What journals work for advertising? Where do you get your sales people?"   And so on. All of that information swapping when there was no standard to go against.

Schachter: Nothing on pricing, of course. [*Laughter*]

Palenski: So long as everyone made his or her own independent decision that was fine, you could talk about whatever you wanted.

Keet: We all made our independent decisions. And everybody had different prices anyway, so just getting the guidelines was all you wanted. But things like distributor agreements: Do you give your distributor 50%, 60%, 40%? Can they make money at 50%? Because we wanted them to make some money. And then we would share what we thought were the cost bases. We weren't trying to be collusive and put together a single policy. We were trying to figure out how to put together the business practices. That's where the real interesting history is in my opinion. There was no model you could follow. You didn't go out there and just adopt the model of an automotive parts manufacturer and say it applied to software. We didn't have anywhere to go. We didn't know what it was going to cost to distribute and support software in Europe. How much money do you have to give to a French company to translate the manuals into French, to provide telephone support and on-site support and to deal with you across the Atlantic? What kind of a percentage split did you have to give them? We felt our way all along on those issues. We were doing exactly the same thing on intellectual property. We didn't know where we were going and we weren't getting a lot of guidance from the other industries out there. Their experiences didn't seem to apply much to us.

Krammer: The question that I want to put to the group is: If someone was looking back fifty years from now, what would you want to tell them?

Goetz: I'd say the interesting point is that there is a healthy independent software industry out there. It's important that there be one fifty years from now. We ended up looking to protect ourselves because there was a large company then, IBM. And there's a large company now, Microsoft. The issues are in a certain sense intellectual property issues. Must they provide interface information or is that a trade secret? Intellectual property issues for software are still out there. They affect the growth of the independent software vendors. There're thousands of them out there. There were thousands in the 1970s and 1980s and the issue is to have a viable software industry.   So I'd say we hope, if we look back fifty years from now, that there will still be a viable software industry not dominated by a couple of companies like, for instance, the auto industry.

Keet: I have a little different cut on it than that, Marty, because I think we are moving in the direction of the auto industry and there will be fewer companies. I think if you went back and looked at the history of the automobile industry, you would find that in the 1920s and early 1930s there were companies selling isinglass windows, and Fisher Bodyworks was a separate company that would build you a custom body on a General Motors chassis. And you had choices for your electrical systems. You didn't have Delco and Remy belonging to General Motors, they were independent companies. The consolidation of that industry was in order to better serve the consumer by giving them one-stop shopping, reliability, a place to go to get everything serviced. I think that's the direction that the software industry is going and I believe that there will be more software under the covers of larger integrated systems and that they will be sold by companies like General Motors and Chrysler.

I think that it would be awfully interesting to see from the standpoint of the evolution of an industry like the automobile industry how that consolidation was affected by companies like Delco and Remy and Fisher Bodyworks. Because they created the opportunity for that consolidation and hence for a very rapid movement forward in technological progress where the technological progress changed from being the invention of the individual component parts to the integration of the component parts into something that became very easy to use. You don't think about it anymore with a car. You get in, you turn the key, you expect it's going to run. You expect it's going to run for a couple hundred thousand miles. You don't really think about needing a radio or whatever the other components are and I believe that's the direction we're going.

Krammer: What would you say, in a few key points, about this early period then?

Characteristics of the Early Period of Software

Keet: Well, it was a period of invention. You know, when I started there weren't compilers. I was there when ALGOL-60 was introduced and it was a miracle. You could actually sit down and write a program in a reasonably high language and get the machine to execute. And then you could take that program to another machine and compile it and have it execute on that other machine.

We can talk about the bad things IBM did, but IBM created the base standard and the technical underpinnings and the marketing umbrella under which many of us flourished. The fact that there was a single architectural standard gave us a great customer base to sell to. The fact that this is a big country with lots of people who all speak the same language, across a fairly homogenous base with common contractual and legal systems was tremendously important.

And so, I would say it was a period where everything was being invented simultaneously. If you had tried to invent an automobile in 1908, you would have had a very, very hard time. You could have built one but you needed people out there to give you all the components. Somebody had to build the transmission and somebody else had to build the motor and somebody else had to put the siding on the thing and somebody gave you the running boards and somebody else built the wheels and that's what we were doing. If any one company had tried to do it all, it couldn't have. I think we're in a phase of consolidation and one of the reasons I think the history is important is because once the industry is consolidated, people will forget, will not realize that it didn't just spring magically out of dust.

At TSI, we wrote a telecommunications monitor system. That was our base stock in trade. You wouldn't think of doing that now.   That has to be part of the underpinnings of the operating system.

Goetz: Maybe at the next session this afternoon we can continue the discussion of whether the automobile industry is the right model. Maybe it gives us a model which the software industry in the next fifty years *shouldn't* look like. It may turn out that it will look like the auto industry but would that be better for the world or would lots of small innovative companies be better? Lee may be right. It may happen but will that necessarily be good for the world or for innovation and invention and progress?

Schachter: I'm concerned that in the direction that we're going, the ability of a Microsoft to be born in an IBM era is not going to happen in the next era. The possibility of another Microsoft being born while Microsoft is so dominant is not going to happen. Lee's scenario is the one that is going to prevail. And that is going to, I think, kill innovation.

Keet: Well, if you believe what was said in one of the speeches this morning, the industry has doubled in eleven years. If I'm remembering my rule of 69's right, it says that the average percentage growth rate is less than 7% and, if we are as an industry growing less than 7% per annum, the innovation and the growth from a financial incentive basis in an entrepreneurial or capitalistic society is going to be consolidation. So I think that there may be an inevitable economic law regardless of whether you're proselytizing on...

Goetz: I'm certainly not anti-General Motors but there was a phrase when GM dominated the automobile industry: What's good for General Motors is good for the world. Unfortunately, it took the Japanese to make GM realize that they better improve their cars and do a lot of other things to compete. So I think competition is important whether it be twenty companies or two thousand.

Keet: I want to add one more point in response to Joan's question There was a lot of stuff that had to be invented back in the 1950s and 1960s and 1970s because it didn't exist and, if you didn't have it, you couldn't build a system. Building systems today is much more a matter of adding bells and whistles. That isn't to say that these bells and whistles aren't good, but they aren't fundamental underpinning stuff that you absolutely have to have and that, to me, is characteristic of a technology era as it matures.

Once the underpinnings are there, consolidation is where consumers see the value. And they pay the bills. So I believe that we're in the consolidation phase of an industry and that fifty years from now software will not be viewed as a separate business. And, that's why we want to preserve its history, because it was an important separate business.

Aspray: Separate from?

Keet: Completely integrated solutions. The example I'll give you is there is more computing power in your automobile today than there was in the entire world thirty years ago. The increase in the processing power in the last thirty years—this is a real number—is thirty-three thousand to one. And the next thirty years is going to be thirty-three thousand to one and you're going to have computers and software everywhere. I mean *everywhere*. It's going to be in your pen, it's going to be in your refrigerator, it's…

Goetz: I'm going to take the other side. Large corporations, small corporations had problems twenty, thirty years ago building applications. They still have problems. The need for better software is still there. What's happening is they decide they're going to outsource because their IT department can't deliver. They outsource and they find out the outsourcer can't deliver. There is still a radical need for more and better software. If you look at the power of hardware, there's no question that it's increasing.

But if you look at the typical large corporation, one that is spending millions and millions of dollars each year, they continue to be in a panic state. They've been in it for twenty or thirty years. They survive, they get by, but no one is really happy with building large software systems, mission-critical systems. And I think that problem may be with us twenty or thirty years from now.

Obtaining Legal Counsel

Aspray: I have a question about expert advice during the 1960s and 1970s. We heard that there weren't many good models in other industries about how to handle these intellectual property issues. So as a software company you went out and hired legal counsel. Where did you go to find them? How good were they? Who were the kinds of people that were good choices and what were the kinds of typical choices?

Goetz: Well, I believe ADR got lucky and happened to find someone at a conference that was into software pretty early. You know, lawyers adept at intellectual property existed long before software.

Aspray: Were intellectual property issues dealt with by your principal internal legal counsel? And if so, what would you look for when you were hiring such a person? Did you look for somebody that came from the traditional computer industry or somebody that came from the Patent Office or just what was the background?

Schachter: If we're talking about the 1960s and even the 1970s, if you wanted to hire someone who was specifically an intellectual property lawyer in the patent or copyright area, there were no lawyers who were trained in software, or there were very few. Since, as Lee said, most of the work needed to be done by your contract, you needed to make sure you had a tight contract which both protected your software and protected you in whatever other way contracts traditionally protect you in terms of making sure you got paid and things of that kind. I would say that most people went out and hired contract lawyers, people who had a background in contracts and business law. Intellectual property really was the domain of a lawyer in a very large software or hardware company. The average software company would go outside for that particular expertise.

Keet: We hired a local law firm. The company was based in Connecticut and we hired the biggest law firm in New England at the time, Cummings & Lockwood, and got a really good contracts guy, Howard "Woody" Knight, who drafted all of our contracts. He was a great lawyer, but he didn't know a lot about patents. He knew enough to know that he didn't know enough to help us and he put us in touch with another firm in Connecticut that specialized in patents. They gave us the advice that we didn't have anything that was patentable. They also gave us copyright advice, but basically we worked through the contracts guy and in the end it was the right answer. The contracts were the way to go.

Then people came along through the trade association, like Ron and others, who knew a lot about these issues. And we then had everybody in touch with everybody else. They'd come to the meetings and would get advice. Everybody started to exchange information, but I would say that it was in the late 1970s when that really got going.

Palenski: Interestingly, this morning I was looking for a tape that Fred Lafer, Milt Wessel, Herb Marx, the people who founded the Computer Law Association, made on that group's 25th anniversary. Actually, that group, which today numbers some 2,000 members from around the world, sprang from meetings at ADAPSO. It was an attempt on the part of the lawyers who were trying to serve this community to meet and to share information and insights.

Goetz: There were many lawyers back in the 1960s and probably the 1950s that were patent, copyright, trade secret lawyers, intellectual property lawyers. They didn't have a lot of experience with software, but they adapted quickly as most lawyers do.

Palenski: There are lawyers out there today who are trying to reinvent themselves as biotech lawyers.

I can't believe it, but our time is up.

Aspray: Thank you all.