An Interview with

LEE KEET

OH 341

Conducted by Philip L. Frana

on

3 May 2002

Washington, DC

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

Lee Keet Interview

3 May 2002

Abstract

In this oral history software entrepreneur Ernest E. (Lee) Keet chronicles his education in programming and engineering at Cornell University, his early work as a salesperson and systems engineer for IBM in White Plains, NY, and his success as founder and CEO of Turnkey Systems, Inc. (TSI). Keet characterizes the heavy ‘craft’ versus the 'art' components of computer programming; he discusses the introduction of TSI’s Task/Master, the first commercial telecommunications monitor; and his entrepreneurial efforts in Europe. Keet also delineates his role as a member of the ADAPSO software protection committee, his founding of venture capital firm Vanguard Atlantic in 1984, and his service as a board member for the Keet Foundation, dedicated to the support of people and organizations in the Adirondacks of New York. Other topics include software product pricing, marketing, and women in the software industry as well as competitive relations with IBM. He describes his work at Dun & Bradstreet (D & B) in establishing and managing their software division and his management buyout of parts of this operation. This oral history was co-sponsored by CBI, through a National Science Foundation grant project, "Building a Future for Software History," and the Software History Center in conjunction with the Center's ADAPSO reunion (3 May 2002).

Preface

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

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

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

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

ADAPSO History Program
Lee Keet Interview

Philip Frana: This is an oral history with Ernest E. (Lee) Keet conducted on May 3rd 2002 in Washington D.C. in conjunction with the SHC-sponsored ADAPSO reunion meeting.

I'd like to start by just asking you some questions about your youth and your experiences. What kinds of things in your educational background and life experiences led you towards the software industry?

BACKGROUND

Lee Keet: I didn't even know that software existed until I was almost a graduate student at Cornell, so it would be hard to say what led me there. I got interested in science and technology early. I was, I won't say gifted in that direction, but I had more interest in the sciences and math. Later in life I discovered that I actually liked literature and history and probably had not followed the right calling. My father was a physician, both grandfathers were physicians, my father's brother, my uncle, was a physician so there was pretty much a tendency towards pushing me in a science direction. I was only sixteen when I left home to go to Cornell. My father, when I was still fifteen, handed me a couple of books and said, take a look and see what course of study you want to follow. And whatever it was he gave me in the medical field, it might have been *Gray's Anatomy* or something, I opened to the wrong page because there was an open torso with intestines hanging out. And then he handed me a book, which I think was the mechanical engineer's handbook. I don't know why he would have had one of those, but it had very lovely-drawn gears and gauges and mechanisms, and it seemed a lot more interesting and less gory so I said, I think I want to be an engineer.

Frana: And you were a tinkerer already?

Keet: I was a tinkerer. I was perhaps a bad boy tinkerer. I was making home made explosives when I was twelve and built a network with a buddy of mine with bell wire that we acquired in 25 ft. sections and then strung together between our houses. Our houses crossed two streets and one intersection, and we strung the wire from lamp poles and neighbor's trees, much to their annoyance. It took us quite awhile because we had to buy the wire by collecting discarded bottles and selling them for the redemption value and using the proceeds from our paper routes. And we were both, I think, twelve at the time. But anyway, we finally got it built and it actually worked for about six months. We could call each other with only two wires but we had it figured out, actually I figured it out, how to put a ringer and two way conversation on two wires. It annoyed his mother so much because my friend Ted would be off riding his bike or something and I'd be ringing him and driving her nuts, so finally she said, take that thing out.

Frana: Did it have a hand crank? Is that how you rung?

Keet: No, no. We used 9-volt batteries. I think it was 9-volt, it might have been bigger, you know the heavier 6-volt. I don't remember on that point, but it was electrically driven and I'd also built a radio when I was a kid so I had some interest in the topic. And it wasn't that I didn't enjoy the math and the science at Cornell, I just wasn't terribly interested in it. I mean it was all exactly what the book my father had shown me. It was gears and mechanisms and it wasn't too much fun, and then I discovered software in my senior year.

INITIAL COMPUTER PROJECTS

There was no computer science program at Cornell at the time but there was a programming course. Of course there were computer studies in the electrical engineering school, but I was in mechanical engineering and the only thing that they had was a programming course which I took in the last half of my senior year, that's my fourth year.

I should give you an aside. Cornell at that time did not issue a bachelor's degree of science and a master's degree of science. They issued one combined degree, which had the strange title of Bachelor of Mechanical Engineering. You had to go five years for it. They have exactly the same course of study still to this day but you get a BS and an MS. But then we were in for the duration of five years. So at the end of my fourth year I had taken this course learning to program in ALGOL and in FORTRAN and I thought this was neat. I liked this stuff. And I also really liked the professors, two guys at Cornell who went on to greater glory – Bill Maxwell and Dick Conway. Also, I had a friend, Pete Giles, who was a half a year ahead of me. He graduated in January of 1962. I graduated in the June class. So Pete was going to be leaving in six months and he convinced me that I should take over his role on the project that they were doing, which was a programming project. It was an attempt, under Bill Maxwell's direction, to do a simulation of a theoretically very difficult process which included inverting a 40 by 40 matrix. A trivial task today, I mean you can do it on your pocket calculator in a couple minutes at worst case. But it took us, on the Burroughs 220 that we were using, all night long. We weren't very skilled. We didn't know about writing programs in segments with checkpoints and restart mechanisms and the input/output devices were so feeble that the program would frequently get to conclusion and it would kill itself trying to spit out the results. And there was no way to store the results easily because the storage was volatile. It was a drum that rotated in a bath of oil and it would keep the data for a while but not forever. So it was very frustrating. We finally got the project done.

Frana: And you were doing this at night because that's when you could get the time?

Keet: Oh yes. We could only get the time from sometime between six and nine in the evening and six in the morning. And frequently it would be even later than that. Every once in a while there would be some other project that would come in. We were only assured of time from midnight until six, and I assure you, we used every minute of that time through the whole spring of 1962. We got the project done. We published it as our master's thesis.

Frana: Together?

Keet: Together, the three of us as a team. And it was successful; I mean we won, believe it or not, we won an award for it – the Silent Crane and Hoist Material Handling Award of 1962 –I just laugh at that title - but it came with a $25 dollar check for each of us,

$75 bucks total. And I was so poor and at that point, Nancy and I had just had a little baby, and that 25 bucks was extremely important money. So that's how I got interested in software.

IBM

Also the summer before my final year I had been able to do some work with IBM at Endicott, NY. It was kind of a release-time program from Cornell and I really loved it on the shop floor. I've always loved shop floors, and I just loved being down in the Endicott plant and seeing what they were doing. And I got interested in IBM as a company and I got interested in the computers that they were building. They had just started to switch Endicott over to building computers from the electronic accounting machines (EAM). So, when it came time to apply for jobs, I put IBM very high on my list and the rest is history. IBM offered me a job in Elmira and I went back to Nancy who had endured now her second winter in Ithaca and - Elmira isn't that far away - she said basically, no, let's not do that. So I called IBM and I said, I'll take your job if it's anywhere in the New York metropolitan region and they said, where would you like? And I said, I don't care, White Plains, New York, Bridgeport, Connecticut, anywhere, as long as it's closer to where our families are because we've got a little kid and the grandparents would like to see the little kid, and if you can find a place I'll take a job with IBM. And I had a competing offer from a company in Waterbury, Connecticut, for more money, but IBM came back and said, we'll give you a job as an Associate Systems Engineer in White Plains, New York, and I said "sold" and off we moved. So, that's how it got started. Of course my first job was not programming, it was wiring panels but that's a different story.

Frana: You got very good at it, partially because you had to come in and rescue other people.

Keet: Well it's kind of a trivial discipline, compared to general programming, it's a real subset because everything runs on control cycles and I guess maybe my mind is just more logical than some other people's minds in sequential logic steps. At the time IBM had hired a bunch of people, mostly young ladies, to wire these panels. They hadn't really selected them for their logical skills. They subsequently came out with a program called the Programmer's Aptitude Test that they used in hiring people. It's actually a very good test. They had to withdraw it because it was deemed to be discriminatory in that it was sort of like an IQ test and therefore, it induced certain problems. I don't know all the details behind why they withdrew it. I used it a lot when I was at IBM for other people after I discovered how good it was. But in any event, back to the main theme: They had these young ladies, they were called, believe it or not, "systems service girls." And they would go out and help the customers install their EAM equipment. They were really lovely ladies but they knew they were on a dead-end career path. There was just no advancement hope for them, and they were there either to find a husband or to make some money and move on. And we were the first group of classes that were hired with a lot of testing. I mean they tested me at Cornell for two days (with three interviews and two tests) before they made me a job offer. So I knew that there was a screening process. I knew they were being very selective and that's what put a more professional systems engineering group together at IBM which subsequently became extremely important for IBM's success because of the explosion of computers. You know, they never expected to sell many computers and when I got there they were still thinking that maybe a thousand IBM 1400-series machines would be a lot. And they ended up selling as I recall 21,000 1401's alone.

Frana: I think the first estimate was 200 or something crazy like that.

Keet: Yes. Well there is a famous quote from Tom Watson Sr. back in the late 1940s where he said he thought that the worldwide demand for computers might be five. I think that's the right quote. But anyway, that's sort of how I got there and then the projects that were assigned to me brought me directly in contact with some of the more advanced programming demands as the computers took off. In 1964 when IBM introduced the System/360 I was in the need-to-know program and was exposed to things that just blew my mind, like operating systems that multitasked. It never even occurred to me that a computer program would do anything except run from start to finish.

PROGRAMMING AS AN ART

Frana: You said that you yourself were a lousy programmer. I find that hard to believe. But what makes a great programmer then, from your perspective? What did these girls not have that they could really have used?

Keet: Well, I had more than the girls, but it's just sort of like saying what makes a great painter or something. You go beyond 'can you draw' to 'are you an artist'? The great programmers are artists. And the discipline that I find is closest to it is being a lawyer and writing a contract. It's very difficult to be that well organized. Any good lawyer will check all the entries and exit points for all the ifs, thens and whereases. But when you're finished you frequently have a piece that works, that hangs together but has no elegance to it. And I'm using that analogy because I think a layman might understand it. If he picked up a document and it was somebody's will or it was a business contract and it was just so obvious and logical that if this happens then this person gets that and if not, or if my wife predeceases me or whatever it is that is being written, you just look at that thing and it's beautifully and elegantly produced. That is what a great program is like. I wouldn't say it's a science but a craft brought to the level of art and it's not trainable.

Frana: That's my next question is; can it be taught to people?

Keet: I don't believe so. No, you have to have the basic skill. You can't be a great artist without having the craft. You have to know how to mix paints, you have to know how to prepare the canvas. You have to know how to put on layers of paint. But that's craft. When you're finished it could be something that you couldn't sell in a garage sale or you could have a Vermeer. Great craft, spectacular craft but not great art. Some modern painters in particular don't know their craft very well and their works are very hard to curate because they are falling apart, but it's still art. I don't know if that analogy works but I've never been able to train programmers beyond the craft. In other words, I can get them to the point where they understand the craft, but I can never get them to the point where they could become the true artists who could code elegantly, flawlessly with the kind of result that somebody else could pick up and understand and work on and maintain. That is almost impossible for me. And I was a lousy programmer because my stuff worked and I did it very fast but it had a lot of Rube Goldberg qualities to it.

Frana: It was functional but it didn't have that level of art?

Keet: Functional, but I'd get 90% through the program and realize that I could have done it better if I had done something else. And then I'd tack something on. I would go back and look at my stuff and it would be clearly the work of somebody who understood the craft but was impatient and was no artist.

Frana: That is very humble of you to say such a thing.

Keet: It's not humble. It's true - it's absolutely true. If I ever went back again and they said, your profession is going to be programmer. I would say, wrong. We used to say programmers clustered around zero and a hundred. They're very few fifties. It's very difficult to be an average programmer. You are either really good, or you're really bad. I was the rare exception, I was a fifty. I mean I just couldn't get to a hundred. But I could make my stuff work.

Frana: Is this why there is this myth in the software industry, true or not, and in computer science, that they really value certain non-traditional routes to expertise, the kinds of experiences that people get? They're not sort of routed through generic educational institutions and then go on and are magnificent programmers. They have to come from some exceptional background, whether good or bad.

Keet: No, I think there's an innate skill, and that's why I think the IBM testing, the programmer's aptitude test worked. It didn't necessarily require a lot of prior education. In fact I used the programmer's aptitude test at a company that was a customer of mine at IBM, the Bullard Machine Tool Works, where I had won a very significant order taking out two Univac machines in the very early days and replacing them with IBM System/360 Model 40s when the System/360 had just come out. This must have been 1965 I would guess and we needed talent to program these machines and there was a dearth of talent in the market. The machine was too new. People hadn't been trained. So I took the programmer's aptitude test and with the CEO Ed Bullard's permission, gave it to every employee of the Bullard Machine Tool Company who was interested. He posted a job opening that said, we're looking for programmers, systems people and potentially managers for a new department and if you're interested IBM will administer a test. I shake in my boots when I think about that now. You need to remember, to put this in context, that I'm all of 24 years old and I'm giving this test to grizzled employees who have been running turret lathes and stocking inventory. Anyway, we found six people out of the entire population who scored extremely well on the programmer's aptitude test and we sent them off to the IBM schools. Three of them came back with the equivalent of straight A's, and those three were promoted into the systems department. And all three of them went on to great careers. One of them went on to become the CIO of United Technologies. Another one went on to be a systems programming manager for a large company. They had no prior education or training and they just leapt off the page as being the kinds of people who grasp it. Bobby Fisher is a good example – the chess player. You know, almost an uneducated guy whose mind works in a chess-like mode.

So I'm reinforcing the theme that the best programmers in the world are going to be programmers who are technically very well educated and from that population you select the ones who have the unusual aptitude because otherwise they may not understand the business processes that they are trying to program. It would be great to have somebody who is doing an accounting project understand accounting. And it would be great to have somebody who is doing an engineering project understand a bit about the mechanics or the physics of what they are trying to program. But out of that group if you said to me which would you rather have, a mechanical person or a physically trained person who has no skills in programming or an uneducated person who has great programming skills. I'd take the latter anytime.

One of the best programmers I think I've ever hired in my life was a guy who, last I saw of him was a politician running a small district in Virginia – an alderman or selectman or something like that. He's no longer writing programs, which I think was a great loss to the world. When Steve, his name was Steve Ward, walked into my office, and presented himself for a programming job, I immediately liked the guy. He had all the right credentials. I had him interview a couple of my other employees. They liked him and said he was terrific. He had only had one programming job before and this was on a Wednesday and I said to him, the assignment is to write a source code generator that will take previously written source code and with some commands inserted in that source code let you customize it. And he said, "Oh that's a trivial project," and I said, "Well there's a problem. It's got to take COBOL source code and generate COBOL, but the program itself has to be written in COBOL because we are going to distribute the whole thing as source code to customers." And he said, "I don't know COBOL." I said, "Well we have a self-study course here," and I pulled the manuals out for him. He said, "Okay, let me at it." And I said, "Well how long do you think it will take?" He said, "Oh, I don't think it's going to be long – can I have the weekend?" I almost laughed out loud but it is a true story that on Monday he brought in the product Genesys that we never sold for a fee but we distributed with every single Task/master, Key/master and Docu/master that Turnkey Systems ever sold. We hardly changed it from that Monday when he delivered it. It was perfect, and anybody could understand it. You could pick it up and clearly read his code. It was one of those masterpieces. And he had never programmed in COBOL before. So, it just illustrates my theory.

Frana: When you discovered you were a 'fifty', then what did you do?

Keet: I hired people like Steve Ward.

Frana: Clearly, you knew you were a 'hundred' in other areas.

Keet: No, actually I wouldn't describe myself a hundred in any area. I would describe myself as a fifty in many areas. I've done all the jobs. I've been the CFO. I've been the CTO. I've been the CEO. I've been the COO. I've been the VP of Marketing and I've been a salesman and a sales manager and I would describe myself as barely adequate in all those jobs. I can do those jobs, but if you sent me out in the open marketplace and said, "Okay Lee, you have to compete for the VP of Marketing's job," I think I would put up a pretty good fight but I don't think I would describe myself as anywhere near the best VP of Marketing that I've ever met and probably on a scale of 1 to 100 I'm a 50.

Frana: So it's like the alternate definition of virtuoso. Not necessarily the master of the violin but good on lots of stringed instruments.

Keet: Jack-of-all-trades and master of none – that's me. And I'm also very impatient. So if I were a virtuoso, to use your words, I'd probably get bored stiff very quickly. I like changing jobs and changing scenery and changing the people I am working with frequently. That doesn't mean every day but it means my attention span in business has tended to be three to five years and then I, for reasons sometimes of my own making, move on. And sometimes it's not of my own making.

TURNKEY SYSTEMS, INC.

Frana: Tell me a little bit about Graphics. That was a very early piece of software.

Keet: Graphics was a product that we actually started to sell in 1969. It wasn't a very good product. I mean it was very clever in the sense that it did the job that we needed done but, you have to understand that we started Turnkey Systems as a custom contract programming house and one of the early projects we got coincided with the release of the IBM System/360 support for the graphics 2260 terminal. The 2260 was a pretty primitive device. It was just a little bit better than the glass teletype. There were a few characters that you could put in to make a field not be type-overable by a user and something to highlight a field and so on. But it didn't have too many options, unlike the subsequent 3270 which had a lot more control characters. The 2260 was a pretty dumb terminal. But it was a terminal and people said to IBM, what can we do with it? And they said, you can write interactive programs for it. Well, it would be fine to say that you wrote a program that controlled one terminal but then if I wanted the same program to service my other terminals, how do we share that program? IBM did not have an answer to the sharing issue. If the program is at all complex, you might want it to go out and read a file. Of course, if two terminals are using the same program, they are both going to want to read the file but would not be doing it in sync. You might be looking up inventory and I might be looking up inventory and then you might be saying I'm taking that inventory and I might be saying I'm taking that inventory and then you might be saying, I took it, and then I'm saying, I took it. And the last one in is the one you see later. You took it, I took it, but it only shows me taking it.

Also there is the question of control. The IBM System/360 was a multi-tasking machine which meant that at the operating system level it would allocate control to protected partitions and if one of them was busy waiting for something to happen, control would pass to another partition. It could then execute to the point where it was waiting for something to happen and the stack of requests were maintained by the operating system so that on an event completion it knew which program to pass control back to. And this is the way that I could have a batch of cards going in the reader and being compiled and I could have a program running in another partition that was printing the payroll checks and every time the program that was printing the payroll checks went out to the printer and said print, because that is a very long-playing event, that program would relinquish control. It was still sitting there- all the pointers and everything right where you want it to come back when the print execution finishes - and the other program would continue. Now assume I take out that print payroll check program and I put an online program to control 2260s. If they ran sequentially, if I came in and said, I'm now going to use that partition, I'm going to ask for a record, get the record from a disk, format that record and display it on the terminal. These are long playing events that can't be interrupted because there are no other programs in my partition; there aren't any – there's only one. So when I go out and I say read a record, the payroll program gets it or the compiler gets it and the terminal control is single thread – it's me and me only. Until I'm finished and my program exits, nobody else uses it. Not good.

IBM had a very primitive routine that they distributed, as I recall as a type III program, which meant that it was IBM produced but not supported. This program allowed you to get at the very lowest level, the interrupts associated with the 2260. We said that what we need is an operating system-like program that runs in the protected partition and allows multiple terminal devices to share that program so when one of them goes out and says send the message to my terminal another user can execute that same code. Without diagrams this is hard to describe, but we invented something called pseudo-conversationality which meant that each user of that partition could execute the same program start to finish and you didn't have to run a different copy of that program in another partition in order to have two users. It still didn't solve the problem of people sharing the code. It wasn't sharable code. It simply meant that only one copy had to be there. That was okay we thought for the year 1969 when interactive applications had very low volumes. And, in fact, it probably was adequate for the early users because most of them had a dozen, two dozen terminals and with low volume you didn't notice that terminal A was reading a record and displaying it to the terminal before terminal B was reading a record and displaying it to the terminal. Sequence was a problem only in terms of performance. It wasn't until much later that hundreds and thousands of devices were trying to be controlled.

Graphics was pseudo-conversational, it didn't do exclusive record control, it didn't have to because it went start to end in the partition, so if I read the record and updated it, so what, no one else is executing that piece of code. So it didn't have to solve that problem. There was only one terminal type to worry about, the 2260, so we didn't have to solve the problem of varying device types because IBM customers started off using 2260's. And we thought that this was great. Well four customers thought it was great too and that's the total number that we sold in 1969.

TASK/MASTER

We immediately realized that you can't do it this way. You need full multi-tasking and multi-threading. You need record control at a minimum so that you can read and write without anybody destroying the database or confusing the database. You need multiple terminal support and that has to be made easy for the user. So we went back to the drawing board and we built all of that into the Task/master program which we introduced in 1970 and which was the most successful telecommunications monitor in the market for a while, for quite a while. And it had some really nifty features. It had something called terminal independent modules or TIMs that allowed you to do something that is very XML-like today which is to describe the way you wanted your message to look, and then the TIM converted it to the 3270 or 2260 or the Teletype or whatever the device was.

You gave the message a meta-description and you did that at a very high level. If it couldn't satisfy your request, for example, if you asked for a blinking field and the device you were sending it to didn't have a blinking field, it would do its best by maybe underlining. In other words, there was a table that would show the user what would happen and the programmer or the developer would say, no I don't want that – that's not good enough. If I can't have a protected field it won't work. But they at least could choose and then they could write one program and they could still use older terminals. And then we did build in multi-tasking which was a very clever piece of design, especially since the whole thing was written in COBOL which astonishes people to this day.

Frana: Now was it entirely customer-driven? I mean was there a specific event that drove you to incorporate multi-tasking?

Keet: We knew all along that we had to do it. But there was a specific customer at American Brands where the manager there must have had time on his hands because he commissioned one of his people to run some sort of a test to see if it actually was multi-tasking.

Frana: And of course he didn't need it.

Keet: He didn't need it. He wasn't having any response time problems. He just was checking us out. Unfortunately we failed the test. So we sent him off a technical bulletin that said that multi-tasking had been disabled in this current release and would be re-implemented in the next release. Then we scurried off to UniRoyal which was the data center we used in Oxford, Connecticut, and we literally lived there. I mean we'd sleep there, eat there in their cafeteria and we built the multi-tasking version in a matter of, I hope I'm remembering this right, but I think it was six weeks.

Frana: That's a good story. Your've recounted it in the *IEEE Annals* in V24:1.

Keet: Yes, and we delivered it to the customer and the customer of course didn't need it but ran the test again. And said, oh good, it's working. Actually we weren't sure it was. We had tested it ourselves, but we didn't have live customer data and they did. But the early software industry was like that. There was a lot of what later became known as vaporware with a lot of promises made in advance of the code. I'm not at all apologetic for that. We were scurrying to build what the customers wanted and if we thought we could get an order although one gizmo that they wanted wasn't in our product, more often than not, we'd tell them that it wasn't there but commit it to them as part of the order. But every once in a while it would be implied that it was already there and then we'd scurry off and build it.

Frana: But you never made promises that you couldn't keep.

Keet: No, and as a matter of fact we never lost a customer that I recall. You know it sounds trite but the customer always came first. And we would work them hard for money. I mean we would work them real hard for money, but we always tried to keep them as happy as we possibly could. And over the years we built up quite a large following. Task/master died an unnatural death in 1978 and it was in the early 1990s that I discovered that there were still quite a few people still using Task/master.

Frana: Why was Task/master written in COBOL?

Keet: The strategic premise that we were operating on was that source code customization at the customer level would give us major advantages. In other words, we could have features that could be selected or deselected, and they – our customers - could do it. Also there was the fact that we were a little company and all our wares would otherwise be hidden. Without COBOL source code, the customer could not be told with any confidence that they could take over and support what we delivered to them. The fact that we were delivering them pure COBOL source code gave them great comfort - they didn't need escrows, they didn't need anything else. The fact that they could customize it gave them even greater confidence so that they could say, you know we are only an OS shop, we only use 3270s. We don't need this feature or that feature. And they just specified that in the generation process and they'd get perfect source code for their own configuration. They could of course add their own source code to it if they wished, and a number of customers did and we gave them exits which meant that when a new version came out their stuff would almost always work with the new version. So it was good for them and it was good for us and we had some designs on eventually being able to port it to other machines. We never did. We only ported it within the IBM line but you have to remember that these were the early days when the competition was still trying to fight the System/360s and COBOL was ubiquitous. You could compile it on almost any machine.

SOFTWARE PIRACY

Frana: When did you get concerned about sharing your source – or did you?

Keet: I never got concerned. One of my early lessons was that customers really respect contracts and in fact written contracts are seldom referred to – they respect the business terms. So if you're doing a business-to-business deal with a big AAA company, the final deal as documented, whether that means contract, memo, letter, or even remembrance of the parties is worth a lot more than anything else. I never had any customer steal anything from me. I had distributors steal from me but I never had a customer steal from me. Our distributor in Munich had been to my house in the US and brought me a bottle of wine with my name on the label that he had specially printed He was a glad-hander. I won't name this man. His wife was a schoolteacher and I knew he was struggling. They were just getting the business put together and he took me to this palatial house in Munich – Munich is a very high-rent city - and then showed me the swimming pool that he had just finished in the basement. I immediately got suspicious. A schoolteacher and an entrepreneur with a mansion? He was running a division of a parent company, and lo and behold it did turn out that he was stealing from the parent company. He had put dozens of his relatives on a fake payroll and was siphoning money into Switzerland, but he was also stealing money from us. He had sold quite a few copies of Task/master that were unreported. One of the customization steps was that every customer's name appeared in the generated source code. We got a note from one customer saying that we don't know why you keep sending us updates that say another customer's name, and we went, "huh?"

Frana: Some flags were raised.

Keet: Right. Right. So, no I never ever had a customer steal, and I never worried about giving source code and I'm sure that some of them could have just said, you know we're not going to make payments and we're not going to pay you any update fees. To my knowledge no one ever did that.

Frana: Now is that where you got the idea for putting in junk code so that you could track it?

Keet: No, we put the customer's number and name in when we generated the source code for them.

Frana: In your book about piracy you say something about this.

Keet: Oh it's easy to trace. If you throw something in you can always trace it. Nobody is going to take the time to do that. But my biggest point would be that business to business is different from business to consumer. Microsoft has an entirely different issue with its fear of piracy than we did when we were selling software to General Motors or Dupont. Dupont was not about to steal from us. In fact, if anything they were being overly scrupulous to make sure they followed all the rules. And they had batteries of lawyers who were attentive to that sort of thing. This difference between consumer software and business software is part of the reason that ADAPSO later lost out to the Software Products Association and BSA who were dedicated to preventing the thievery of mass-produced software. There just wasn't the need in a corporate to corporate relationship, and ADAPSO missed that point. That was much more of interest to those early software publishers than it was to the early software product companies and I made that distinction in publishing software, which I did later in my career. I made an early attempt at application software publishing in 1983 when we built the product that I funded called SoftTax, but that is not in the mainstream of what you're talking about.

Frana: Yes, we can always reorganize the information but that was an issue that did come up last night. I don't remember if it was Ed Bride or someone else mentioned that avoiding that issue of what do we do about piracy was one of the things that made ADAPSO seem irrelevant. That's how you remember it too?

SOFTWARE PROTECTION

Keet: ADAPSO focused on the large-scale software and secondarily on the laws that we needed to protect us beyond the contract. And our focus came down to copyright. So I wouldn't say we missed that boat. We did an awful lot of work in the area of copyright and in fact we had a lot of help from the people in the software publishing field. Jon Shirley who was then president of Microsoft was on my committee, the software protection committee. Microsoft said, yes, we need copyright. But they needed more than copyright. They also needed physical protection. They needed police. They needed governments, they needed raids, they needed people to go in and shut down bootleggers. So they needed an organization that was willing to go do that with them and for them and ADAPSO was never willing to do that.

Frana: An enforcement organization?

Keet: Well, a means to put pressure on the Singapore government to go out and shut down people who were minting fresh copies of Windows. ADAPSO was not willing to do that. In fact ADAPSO's lobbying efforts were focused on large-scale issues in Washington. And they did some of it in Washington for the Microsofts of this world but they didn't do as much as the other organizations did which is why ultimately Microsoft stayed in ADAPSO but put a lot of their efforts into the people who were really out there to club the counterfeiters. We were much more interested and I was much more interested, in making sure the copyright law extended to software. And what we wanted and the bill that was sponsored was simply to get software added as one of the listed protected arts. It turned out that that was unnecessary. We got it on the floor of the Congress, but we never got it passed. But in the interim there were so many court cases that were going on around the country that it became very clear that the courts were going to take a very aggressive view of a work of authorship and that as software was a work of authorship we were indeed going to enjoy the protections of copyright. But it was a time when a lot of it was groping in the dark. We didn't know which way things were going to go. We didn't know whether there was going to be introduction of special legislation because there were a number of very prominent people who were saying that software is unique, it's *sui generis*, and you've got to get special legislation. I always felt that it was not unique at all, since a piano roll was basically software, and there had been a lot of cases on that and we could just ride on the coattails of other works of authorship if we played our cards right. But there were people who were pressing for patent protection on the basis that it was a machine component.

Frana: Like Marty Goetz.

Keet: Like Marty. Marty was the leading proponent of that and he actually got a patent. Now, even though there are thousands and thousands, tens of thousands of patents for software, I still think they are all invalid. I don't think there is a single possible valid software patent. And I don't believe that the way the patent act is constructed extends to software.

Frana: Did IBM open the door for some of that by not being concerned about it until the mid 1980s?

Keet: I think IBM reacted more to the Japanese. If you look, I think Hitachi and Fujitsu have published as many or maybe more software patents than IBM. I think what happened is that the patent office was so deluged that it just relaxed standards and said, fine we'll issue patents. None of them have ever been challenged at the Supreme Court level and I would be very interested to find out what happens when they are. Of course there's a natural tendency not to get them to that level because people are swapping patents in order to get their protection. It has a tendency to make big businesses bigger because they are the ones that have the portfolio of patents and they are the ones that can trade. Later in my career, after I started Vanguard Atlantic, one of our investment banking clients had failed with one of the first hand held devices that could recognize handwriting. It was an 11 million dollar investment that was going down the drain. It was just too early, the devices were just too slow and clunky and I advised them to sell the patent. To my mind the patent was ridiculous. I mean if you tapped on the screen it opened a bigger window so you had more space to write in, and then you tapped again and you closed it. That was the patent. But we managed to sell that patent to every single hand-held device manufacturer back when everybody thought they were going to be in that business. Go, Tandy, IBM, Microsoft, they all bought it and it was a quarter million dollar license fee and they all wanted it just so they wouldn't subsequently have a product line that was infringing. Anyway, I'm very anti-patent for software. But unfortunately that is the way the game is being played in the year 2002.

ADAPSO

Frana: How far back do you go in ADAPSO and when did you become involved?

Keet: You know I've been asked that question and I'm not 100% sure when I first got involved. I think it must have been right around 1971.

Frana: Do you remember much of the banking controversy? I guess ADAPSO really wanted them to get out of the software business altogether because they were breaking the bounds of their industry by providing these services.

Keet: I remember a little of that. It wasn't in our sphere of activity. A lot of people I knew were fomenting against that. I got involved in ADAPSO simply because some of the other people that I started to recognize in the industry were there and a lot of that had to do with Larry Welke bringing in the software industry group into ADAPSO. I was going to ask when that happened. Somebody here is going to know that, what year that was, but it was immediately after that that I got much more involved.

Frana: It's probably 1969 or 1970; it's probably just before you joined.

Keet: If it were 1969-1970, then it would be probably 1971 when I got really interested and active.

Frana: Do you know why Welke's group joined ADAPSO?

Keet: Yes. I think they didn't have any other place to go where they could get any power.

Frana: And even there they were kind of the odd man out.

Keet: Oh definitely, definitely. But you have to remember that in 1970 you could count the number of companies that were really software companies on the fingers of both hands, so there was no way they were going to get economic or political clout alone. That's a question that you would have to put to Larry and those early pioneers who were in that group. I wasn't one of the ones in that group. We did have a software products company in 1969 which the remnants of the grandchild of still exists. It's one of only about five companies that trace their ancestry back into the pre-1970s era. That's Mercator Software which we bought out of Dun & Bradstreet and which had in turn earlier bought Turnkey Systems which was my company, so it goes all the way back to Graphics in 1969.

Frana: That is really a remarkable pedigree.

Keet: It is, but there are others out there that have been around even in more pure form for just as long. Cincom Systems is the most pure example I can think of because Tom Nies has been CEO forever.

Frana: Now, on what committees did you serve in ADAPSO? The intellectual property committee was one?

Keet: Yes, intellectual property. I was a member of the contracts committee for a short period of time. I worked with Esther Roditti on a couple of the contracts that were of great interest to me. I served on the financial standards committee with two leaders – Jim Porter and later John Imlay – when FASB was promulgating its rules on both software revenue recognition and on the capitalization issue which was a big, big debate for years. There were those who wanted to capitalize everything and those who thought that you should expense everything and they were friends. We weren't fighting as an industry. We were fighting with each other in many cases.

Frana: How did they create that kind of magic, I mean you had some fierce competitors sitting at the same table sharing advice and information?

Keet: Well, a common enemy was a very helpful thing. I don't think trade associations work too well unless there is an enemy to fight and it was clearly IBM in those days. IBM was just a runaway steamroller. They didn't know what they were doing. I have great memories of my days at IBM and very great respect for the people, but they were the enemy then.

Frana: You had a good relationship with them?

Keet: Oh excellent, excellent. Many of my friends were still working there. I still thought highly of them and yet I thought that they were out of control. They were competing at the field level in a way that was extremely destructive to our fragile little industry. So we did everything we could to beat them. Of course Larry Welke helped a lot because he would write software reviews. ICP would profile software products in their magazine and they were, I want to say this generously, because Larry really helped build the industry, they were like *New Yorker* restaurant reviews. If you can't say a good thing, don't write the review. So, they were all good and of course Larry wasn't writing reviews for IBM's products, he was writing reviews for the independent products.

MARKETING TASK/MASTER

We actually had a publication that we could hand to a customer that showed Task/master off. It said the things we wanted to say to them, and we took that to another degree. This was maybe where I hit 100 on marketing even thought I'm usually a 50. I created a checklist of all the things that you could use to compare Task/master to CICS and it was about 8 pages long. It was a booklet. It wasn't just 4 or 5 things, it was eight 8 1/2 by 11 pages, not densely spaced type, but not loosely typed; it was a checklist of some 200 things you had to compare product to product on, and of course we had put in a score and a weight. So it was check, score, and weight.

Frana: You chose the weight.

Keet: No, we left it blank. We left the weight blank and the customer could fill it out but of course the way the questions were asked there was no way that you could even put a YES in the CICS columns so it looked like there was 200 things that we did that they didn't do and then the customer would put its own weight on it and of course we were going to win and then they couldn't say, well that's because you put the wrong weight. Well, that was clever, but we even got more clever because we got a magazine interviewer who really wanted to be either home with his kids or out having a beer or something and he took it and asked us to fill it out. We did, and we put our own weights and he then published it under his byline saying here is a comparison of Task/master and CICS on the important issues using the weights that we had given to them without saying so and then tallied the final score. Well, the magazine was, as was typical those days, was struggling, so we said, "Can we buy reprints?" And they said, "Oh sure," and they printed us a foldout with the cover of the magazine – of course it was on the cover "Task/master versus CICS," and inside was the article and the checklist. And we had that bound with the blank checklist and we would give it to a customer and say, "This is what this magazine says, you don't have to follow their weights or their answers, but please before you buy CICS make sure you do this." And it was great. It was just great. We were fighting for the scraps anyway because there were so many shops that would buy only IBM as a matter of policy. So we were fighting only for the ones that were left over. By the time it got to the mid 1970s, all the database guys, the Marty Goetzes and the John Culinanes and the John Maguires, all had their own telecommunications monitors. So the guy who was anti-IBM might buy it all from Marty Goetz; and our claim to fame was that you could still choose IBM and you didn't have to commit both your database decision and your communications decision to an independent. You could just pick and choose because we worked with them all. We worked with Marty's product and we worked with IBM's product and so we did pretty well. It was a very nice launch to a software products business.

Frana: Now how typical or atypical was that to work both sides like that?

Keet: I think we were pretty unique and of course the application guys didn't have that problem, but when you are talking about system software you get pretty well wired into it and that's one of the reasons that our product, because it was written at a very high level, it was pretty easy for us to make it work with everybody's stuff.

Frana: Did you have special pricing strategies or a user group?

Keet: Yes, we had a user group for Task/master that we started I believe in 1973.

Frana: And did you own that user group?

Keet: It was quasi-independent. We did all the organizational work. We would hire the hall, we would collect the money for them and do the administrative work, but they had the treasurer. They had the elected officials. We didn't spend their money. In other words they would nominally approve everything before we did it but most of them, you know, they would say we're going to have it in Washington and we'd come back to them with some quotes and then my marketing people would do the work. And it was a great way to sell. I think most of us in the industry used our user groups for their benefit but also for our benefit. We would get them to say nice things about us and they would get extra back scratching.

PRODUCT PRICING

Frana: So you get those references. Did you ever do anything like borrowing from the future – future sales. You know, licensing future enhancements to get money today?

Keet: I'm not ashamed to say that we dealt with a little vaporware. You know we would sell stuff that wasn't completely developed. I always looked at it as a way of financing our development. There was no source of funding for any of the software companies in the late 1960s or early 1970s. You only had two sources of money – your own or the customers, and most of us didn't have a lot of money going into this so you really needed to get the customer to pay for things. We would do whatever we could to get the customer revenues as fast as we could. Doing that while you are fighting the tiger was very difficult you know, because IBM was renting their software. The customer would say, "Well, I want to rent the software from you. But we needed cash. So if we were selling a $25,000 product we wanted $25,000 at the time of the sale. And I solved that problem by creating a "hell or high water" lease for software, which is the only reason Turnkey Systems survived. We just didn't have any sources of money other than this. We got the customers to sign what was typically a three-year lease that would either renew at the end or the customer had the right to buy a perpetual license at the end. We would typically divide the price of the product by thirty to get the monthly lease price so it might have been say 800 bucks a month to rent Task/master. The lease would separate out our obligations to service it, fix bugs and everything else. There would be two documents, one would be the warranty and service contract and the other one would be the lease contract; and the lease contract said that the customer would not assert any defense, any claims, would not offset any payments, blah, blah, blah, blah, the whole hell and high water language. And then we took that, and this was really difficult, we took that to bank after bank after bank, all the New York banks, all the Connecticut banks. And you know, we finally found a bank that would underwrite those for us, and they gave us a percentage. I believe we got it up to 80% of the face value of those leases. So you take that $800 a month lease for 36 months times 80%. That's how much we finally got of the total. Well, that turned out to be pretty close in the end to the purchase price.

In the beginning we didn't get that much. We still had to finance the difference. So we would continue to sell our bodies to do custom programming around installation work and everything else. We had a service revenue stream that was essential to keep the business alive while we were building up the products business. Over time of course those product contracts started to renew and the gross margins on the renewals were unbelievably good. And prices were always going up so you got a double whammy with an increased price for customers who did want a perpetual license, because the prices went up and the maintenance as it was called in those days was based on the then current purchase price, our maintenance prices went up. So, over time, this became a very nice sinecure. It didn't get to be self-sustaining, however, for the first seven or eight years that we were in the business. We were still always scraping for money. Fortunately we had some service revenues that kept us alive, together with the bank financing of the hell and high water leases. To say we were cash neutral would be stretching it, but we were close. We weren't taking salaries beyond the absolute minimum and we almost missed numerous payrolls by a flash of the eye. We borrowed money from friends and family, but I think that was very typical. I think you'll hear that over and over again from the early pioneers, that there just wasn't any financing. Nothing.

FINANCING SOFTWARE COMPANIES

Frana: There wasn't any venture capital, there weren't any angel investors?

Keet: No, no, the venture world was also very primitive in those days. Until ERISA changed its rules, money didn't start flowing into the VC's coffers. It was a combination of the change in the ERISA rules in 1979 and the demonstration that venture capital, in modern portfolio theory, as a component of the fully diversified investment pool was a very good thing to have, and the VC's started to demonstrate real returns. The guys who started early funds did extremely well. Stu Greenfield started the Oak Fund and before that had started the Sprout Fund so he was a pioneer. Anyway, Stu sat on our board for four years while trying to raise $25 million dollars for the first Oak Fund, and this was for a guy who was a successful VC. So back in those days it wasn't just the companies that were having trouble getting money from VCs. VCs were having trouble getting their funds started. That all changed of course later so we're talking about night and day between then and the 1990s for example.

Frana: Were there any early spectacular failures that sort of poisoned things for a while or some people who lost their shirt and that turned people off for a while?

Keet: A lot of people lost their shirt but almost all the investors were friends and family and angels. There weren't any funds available. Again you should ask John Cullinane this question. He was the first to raise money on Wall Street. He raised a half a million bucks as I recall. Not a lot of money, even then, for funding a company.

Frana: We're going to go to Boston to do an interview with him. I think that's a very good idea. You always wanted to be another Cullinet?

Keet: We didn't want to be another Cullinet.

Frana: I've got a quote here that says you set a goal of being another Cullinet.

Keet: Well, maybe when Cullinet had become Cullinet and had demonstrated branding. But in the early days when we were just scraping by, John was just another gunslinger on the street. I didn't want to be Tom Nies. I didn't want to be John Cullinane. I didn't even want to be Marty Goetz. I didn't want to be John Maguire. I mean I was perfectly happy being Lee Keet and we were all doing the same thing. And to your earlier question, we were really happy to share stuff with each other. The idea that somebody was going to benefit at our expense of course entered into our thinking but we benefited more by getting feedback. If it had been a one-way street we never would have done it. But to tell somebody what kind of sales commission programs you had put in place and what you expected from your sales people – that was extremely useful because otherwise the sales guys were running you rather than you running the sales guys. Same thing was true with distributor agreements; what kind of terms do you give those guys, because you knew that if you were giving the distributor 50% and another guy was giving him 60%, it wouldn't necessarily be another guy selling your products, it would be a different product line. The distributor would be going towards the profits. So we'd share experiences across the spectrum of problems. A lot of it was just about how you do it – accounting type stuff, contracting. It was always productive to go to the early ADAPSO meetings because there would be people there who would get up and be willing to share what they had done.

Frana: Was ADAPSO sort of mildly collusive as a trade organization?

Keet: I wouldn't say collusive. We were probably insensitive.

Frana: You had a very good attorney I'm sure that kept you honest.

Keet: Milt Wessel was a gem. Milt was a very nice guy, salt of the earth, but he was also a superb lawyer. He'd written I think ten books. He knew his subject inside/out. He taught at Yale. He was a bit of a stick in the mud when it came to doing it right and people would think him a little pedantic. And he was, but he was there and he was on top of the case. We had one meeting where one of our peers got up and started to say, "The only thing wrong with this industry is that we charge too little and we all should raise our prices right now. Let's make a pact, we're all going to raise our prices 50%." And Milt almost had a stroke. He was sitting in the back of the room and he was sputtering. He couldn't even get out any words and we were trying to get the microphone – it was an open mike type exchange. We were trying to get the mike away, and yelling at the speaker saying you can't talk about that because we knew enough then to know that that was dangerous territory.

We probably did other things, even now I don't know. I mean I was just saying things about sales compensation. It's hard to call that one. If everybody is agreeing on the sales compensation, I would suspect that you're in trouble but I've never researched that. We were freely telling each other what compensation programs we were paying – what we expected a guy to make, what would he make at quota, what would he make if he made 200% of quota. You knew what were the base salaries that we were using. And frequently it didn't apply because the product prices were different or the territories were different but it was very, very useful to factor in.

Frana: Well, you can't play the game unless you've agreed on the rules.

Keet: Yes, that's exactly right, and there wasn't a lot of inter-company competition. It occurred and when it occurred it was sometimes painful and sometimes really humorous. But most of the time it was with IBM. It was the customer's inability to break those shackles and say, "Yes, I want to buy a better database product or I want to buy a better communications monitor," or a better whatever.

INDUSTRY GROWTH

Frana: Were there people in ADAPSO that thought of the independent software industry as a zero sum game and that it wasn't going to get much larger and that the only thing you could do was to consolidate and push other people out?

Keet: No.

Frana: Everybody thought it would grow?

Keet: Yes, I think everybody knew in their fiber that it was going to grow. And that's why people like Larry Welke and what they did to get it going so that the rest of the world knew about it needs to be remembered. Larry created the opportunity for people outside of the industry, through its million dollar awards and the ICP quarterly, to see that this was a business. When we started, my folks would say, you're doing what? Software? My first business cards came back from the printer – 'softwear' – And I'm not making that up. I know a lot of people who said that they thought it was softwear. In my case they actually came from the printer – softwear. And I didn't have enough money to have them reprinted so I used the ones that said softwear for six months until we could afford another print run.

Frana: You were way ahead of the curve. Now we've got intelligent clothing now with the implanted chips that you can wear so you should have just kept that title.

Keet: Yes, I should have kept it. I used to say, "I'm not in bras and girdles."

Frana: In doing some research about software in the early 1970s, a lot of people in the popular media were worried about this kind of automation causing people to lose their jobs. That has totally evaporated. Now it seems that software creates jobs. How did that happen?

Keet: Oh boy, that's a complex question.

Frana: It's a big one.

Keet: I think you've got to remember you had the Orwellian vision that was the one that the public had, that automation, Big Brother, was going to come along and would replace the human laborer. And that the human laborer would get nothing out of that so there was a disconnect. If our society is making more things cheaper and selling them for the same price there is a benefit and we will share in that. Well, I don't want to get onto a political soapbox but that is not necessarily true. We've gotten a lot richer but not everybody has gotten a lot richer, and we have a problem that we still have to address as to how we distribute the fruits of that productivity. But I think the intelligent people who were at the time fearful that they would be impacted came to understand that these productivity gains were to their benefit, not to their detriment. Even the workers. Any skilled worker eventually realized that here is an opportunity for me to upgrade my skills, to do more, to be more productive. There is still a lot of fear down at the level of the unskilled laborer and there clearly should be, because we're not there yet. We've got an awful lot more to do in terms of replacing the kinds of work that nobody should do. The question is when you take away the work that nobody should do but people are doing: How do you feed them? How do you clothe them? How do you house them?

KEY/MASTER

Frana: Did you have customers that were concerned about that, “Gee, I don’t want to get rid of my secretary,” or “I need these people for other reasons?”

Keet: We had big discussions with customers about that but I don’t think it ever impacted a sale. We, of course, had labor issues with the Key/master product. Key/master was a product that sat on top of Task/master and allowed you to key enter data sort of the way you would key punch, key to disc or key to tape data, but it was directly key to the mainframe. Because we could now share multiple devices in a single program executing in a single partition under a multitasking computer it was actually a very efficient, cheap and direct way to do things. And in addition you could check things. So, unlike key to disc or key to tape, there could be an exit routine that went out and actually looked and saw if that was a valid part number that was being keyed or saw if this was a real customer or not. So, it was not just key entry, it was key entry with real verification while it was being entered. It was still batch data, it was still going to go through that old-fashioned batch process. It wasn’t being processed real time online. But it was being verified against real time databases so that when it did go to the batch processes there were few if any rejects. Now, these were ladies who did the keying; I say ladies not in a sexist way, they were really all *ladies*. I assure you. I was in hundreds of customer accounts and this was women’s work. They would sit there and they would have a stack of documents on the left side, they would have a keyboard and a terminal in front of them and they would process those documents so fast – it was unbelievable. We did have issues with those folks. They didn’t like it when the computer drove them, they wanted to drive the computer so if the machine was always done faster than they were, they’d start to quit or they would complain or they would go to their union. And we actually had to change the software to accommodate them. So that, for example, we never tried to anticipate their needs. We let them ask for the next thing. We didn’t give them the next screen to enter until they pressed a key for it because if it was there and waiting they felt like it was trying to drive them. And even though it added a keystroke for them to actually get the next screen, they were much happier when that happened. The other thing that was interesting was it was silent and they didn’t like that.

Frana: They wanted to hear the keystrokes?

Keet: No, they couldn’t gossip because the way that they would work is they would sit there and without even thinking type this data in and be talking to Suzie next door and saying that you know Carol is a real mess today. And because they were originally doing it on machines that were noisy like keypunches, Carol couldn’t hear. So in came the 3270s or the 2260s and it was pretty silent and they couldn’t do that. We also had physical issues of design where we actually helped our customers – we told them you can’t put these people side by side the way you did in a key punch room. You either put barriers up or give them some more privacy. There were a lot of physical issues that we never anticipated.

WOMEN IN THE SOFTWARE INDUSTRY

Frana: That is a lot to chew on. We should follow up on that more. What was it like to be a woman in that industry in the 1970s? I mean not as a key entry operator necessarily?

Keet: Well, I hired a lot of women and a lot of them went on to do a lot of good things. In fact the last CEO of Mercator is the person we put in when we spun it out of D&B with the 1985 leveraged buyout; Connie Galley was one of the few women executives in the software industry. They numbered in the dozens but not in the hundreds. But back in the early 1970s I actually got an award from the state of Connecticut for having the highest percentage of women executives of any small company in Connecticut; I forget how they classified small, I think it was under 50 employees. And that's what we were – we had about 48 employees at the time. But it was pretty open for women. I would say the societal changes have affected us a lot more than the industry changes. I think any woman entering the business in the early 1970s could have done anything a man did because we were a later generation. We didn't have the shackles of the banking industry or the manufacturing industry or the insurance industry or hospitals or anything else. We were really creating a new business and even though I'm no longer a young man, we came out of a new generation. Our parents were the World War II generation and the society had changed with the civil rights movement. And I don't remember too much sexism, very little in fact. Just an anecdote. After I left the White Plains office at IBM in 1963 I was assigned to Advanced Application Development which was initially in the eastern region of IBM. Each region had Advanced Application Development. The guys I was working with, and I use 'guy' as a gender-neutral term, included Charlotte Scott, who was one of the real pioneers of the software industry. Charlotte authored Mapics/Copics and all that stuff. I got up as she came into the room. I hadn't yet met all my peers. And the guy sitting next to me (who in this case was a man) literally pulled on my suit coat. Of course we were all wearing blue suits and white shirts, and he pulled me back down into the chair and said, she's fighting for the same job you are. I think there was a general recognition at that point in time that women were equal. So I don't think there was any barrier. I think the barrier has been more self-imposed than anything else. I think that it's societal and we're still trying to seek that balance.

Frana: And that was true inside and outside of IBM. There wasn't sort of an IBM glass ceiling that pushed people into the independent software industry?

Keet: Well, I can't attest to a glass ceiling at IBM in the higher reaches. You have to remember that I worked for IBM for five years and the highest level I reached, depending on how you count, was either senior systems engineer or salesman. I never really got to be a manager. When I announced that I was going to start Turnkey Systems, then all the offers started to tumble out, they wanted me to come back and be a manager. But I had changed careers from being an engineer to being a salesman of my own volition and so I never really would know whether in Armonk there was a barrier. I would say that the fact that Jean Sammet and Charlotte Scott and others of their ilk never really made it into the very high reaches of IBM might suggest that there was, but those are questions I'd ask IBM.

RELATIONS WITH IBM

Frana: You had a very good relationship with IBM, in fact it still bewilders me a little bit how in 1978 you could sort of walk away from the dispute you had with them over CICS.

Keet: I wouldn't make that out to be friendly. I think that was just good business judgment.

Frana: You just really did not want to sue?

Keet: I felt that we could have taken one of three paths. We could have sued. Then there was a less than hidden suggestion that we might get an IBM contract to do custom development work for them as a sort of settlement. Throw us a bone. Or we could have just ignored it. Of those alternatives I thought the best business decision was to ignore it. And it wasn't because I thought they were good guys or anything else. It was because I couldn't see my fragile management team consumed in a suit and taking their eye off the ball. We were just releasing additional products. Thanks to those software leases, as I called them at the time, we had a recurring revenue stream so while they killed the product, they didn't kill the revenue stream completely. They killed the new sales, but they didn't kill the customer revenue stream from Task/master. That went on and declined very slowly if at all and therefore the question was can we make up in new sales from other products what we have just lost in new sales of Task/master. And we hadn't offered CICS compatible Key/Masters or Docu/Masters so we said, here we go guys – they want to take our business, but we're going to go sell to their customers and that's what we did. We kept the Task/master revenue stream and we went to sell more of the newer products to their customers. And I think in retrospect it was a great decision, but I have to tell you we agonized over it and maybe I've misrepresented it somehow but I didn't do it because I thought Ed Kane was a nice guy or Grant Leschin was a nice guy. I did it because I thought it was a good business decision. However, I did think Ed Kane was a good guy, an honorable guy.

Frana: You thought he was a good guy, but that wasn't the point.

Keet: A good and honorable guy. And I didn't think that the guys that I was dealing with at IBM were out to screw me. They thought they made a mistake. I thought they made a mistake. The guys in the field – different story – the salesman trying to get a sale at an account was not always scrupulous, honest, honorable, trustworthy. He wasn't a boy scout. But the guys that were working in White Plains, in Armonk and so on, mostly the guys that I met, were generally trying to represent IBM as the white hat guys. Our decision was a business decision and I think it was a good one and in retrospect I would do it again. But I was fortunate, I have to emphasize, I was very fortunate that the timing was perfect. By 1978 we had built up enough of a recurring revenue stream and a new product set that we could afford to make that decision. Had it happened two years earlier I would have had no choice except to sue IBM because otherwise we would have been out of business or taken the poisoned bone, I mean we would have ended up with a contract doing business with IBM and no guarantee when that was going to end. And when it ended you were out of business because you had only one customer. So, that was never a serious consideration. It was sue or walk away, and we walked away.

Frana: Thanks, that helps me a lot. You really did gut it out, did a cost benefit analysis on that and made the decision.

Keet: It was tough. We hated losing that product.

INTERNATIONAL BUSINESS OPPORTUNITIES

Frana: Now you have a lot of experience in overseas software markets as well. You lived over in England for a while, and in France.

Keet: Well, actually I didn't live in France then. I lived in France later in a different life. I lived in France in my VC life and did start a Pan-European company there but that was in 1989, way after the era we're talking about. I did believe that 40-50% of the revenues of American software companies should come from overseas, just looking at the demographics of where computers were being sold. I was a little naïve about that in the early 1970s not realizing how hard it is to go against fragmented markets with different languages and cultures. In 1975 I moved to London to set up an international distribution center. I turned the reins of the software side of our business over to Dennis Sisco who worked for me and set up in their time zone, or at least within one hour of their time zone, support for all our European operations. The thing that triggered this decision was that Hoskyns Systems which had been our distributor in London, England, decided to exit the distribution business. They were building their own software products and didn't want this annoyance. So I decided this was the perfect opportunity to set up our own technical support center closer to the European time zones. Secondarily, and probably more important, I wanted to isolate that support function from the day to day demands on our US staff because the distributors were always last in line. A customer in Mississippi would come way ahead of a customer in Paris. And thirdly to learn a little bit more about how we could really make this into a worldwide operation. And it was fun. It was a very, very educational time.

Frana: What did you learn? There was an anonymous author at *Datamation* whom I later met who said that European programmers are very different from American programmers. I guess that the European software industry is very different from the American software industry. But in your manuscript you talk about there being some class issues that don't appear, or are not so obvious in America. For example, you say you couldn't have an Australian selling software in England because it's perceived as inappropriate.

Keet: You couldn't then.

Frana: At that point, in the late 1970s?

Keet: Well actually it was the mid 1970s, 1974-75

Frana: Did you find yourself dealing with people who just didn't see software and programming the way you had all your life?

Keet: That's another interesting question and quite complex. Cultural differences definitely enter into the process. We were talking before about where the craft turns into art.

Americans for a variety of reasons, all of which you'd need a sociologist to explore, have dominated the world's entrepreneurial scene. They have been willing to take more risks. A lot of it has to do with our social structure, the fact that you can fail in the United States and then pick yourself up and start all over again. I mean William Zeckendorf Sr. has always been my hero because he built Manhattan or a large portion of it and then went bankrupt. And then because he was in exile and couldn't get any money, he went and built Montreal and went bankrupt, and by then New York had forgotten so he came back and built his third fortune and either fortunately or unfortunately died a very rich man.

INTERNATIONAL DIFFERENCES

Frana: This is the land of second, third and fourth chances.

Keet: And some guys that don't deserve it get it but the difference is that in France not only are you looked at as an idiot, or were in those days, although it's changing fairly rapidly,. but in the 1970s your family, your friends, your schoolmates, your cousins would all think you were an imbecile to go out and take a risk if you had a good education and a career. You couldn't just chuck it all and go take a risk and start a company. Well, that's why in my opinion, the software industry was born in the US because every entrepreneurial advantage you could want except funding was there in the 1960s and 1970s. And I think the thing that is changing it in Europe or was changing it in Europe was the availability of immense funding in the 1990s explosion which did suck people out. I mean at some point they could say, listen there are VCs here with a billion dollars and they are willing to fund my idea and I'd be crazy not to do it. Whereas back then when there was no funding it just was one more nail in the coffin. So, yes there were those differences but I think it did extend all the way down to programming. I don't think that you got that freeness or willingness to attempt things. It's where the craft becomes art. It wasn't in the makeup. You do it the German way and you could build perfect clocks but programming isn't building a clock. You can't follow the rulebook. It's not a Chippendale chair and it's not a Swiss watch, it's a different thing. And Americans for some reason, maybe it's because of the freedom, also maybe because we got an early start and saw that we could do it, were the top of the game. It's changing. It's changing very rapidly.

Frana: Was that cross-fertilization helpful? Do you have anecdotes of things you learned in Europe that became very helpful to you in America or vice-versa?

Keet: Not on the programming front, but certainly on the business side of things. I'm still learning these lessons. I definitely am an American and although I have spent seven years out of the last twenty living in Europe, I'm still learning those lessons. While it's wonderful what we have, the freedom, the ability to fail and so on, a lot of what Europe has is wonderful too and it's very difficult. It's going to be very, very difficult for those conflicts to get reconciled, especially the one that is very troubling to the Europeans right now is the domination of the American culture, especially the bad side of our culture.

Frana: The pop end.

Keet: The pop end as a culture and it's not necessarily McDonald's. You know they queue up for McDonald's. But it's what McDonald's visually represents and how it's changing the landscape versus whether McDonald’s makes good hamburgers or not. I mean for crying out loud they had Wimpies in England before they had McDonald’s and if there is anything more horrible than a Wimpie, I’ve never tasted it. But what I learned and I think I brought back was a much greater respect for perfectly legitimate business practices that you just don’t happen to think are the best. And that you have to work with them. In other words if their DSO, the days of sales outstanding for your receivables, is typically 120 days in France then you just can’t force an American model that says 70 to 80 days is the proper DSO down their throat because you will just lose. I mean you are not going to change their business practices enough to make that work. You’re going to have to find the financing so that you can collect your receivables on their cycles. And we fought that for a long time and didn’t win.

ADAPSO VERSUS ITAA

Frana: We have just a little time left but I wanted to ask you just a couple of questions about ADAPSO and ITAA; were the cultures of those two organizations very different?

Keet: Well ADAPSO became ITAA gradually. It didn’t happen suddenly. What happened was that in the beginning all of us who were running companies wanted to be there for a variety of reasons, policy decisions related to how we were going to deal with the common enemy. We didn’t want second lieutenants doing it. We were learning things that related to the strategies of our business. I wanted to hear it from John Maguire or Marty Goetz or somebody. I didn’t want to hear it from Marty’s sixth level marketing guy. So there were real strategic reasons why we wanted to assemble and then that became self-fulfilling and became a social thing. We got to like and respect each other and our wives got to know each other and we’d have dinner together and we’d come half socially and half to learn and that sustained itself really after the enemy wasn’t the big issue anymore. It sustained itself way beyond where you would have expected it to simply because the social gathering itself became the reason. But at the same time ADAPSO started to attract, because the battles were over, the enemy under its covers. So IBM was now a big member, AT&T was a member, Nynex was a member, and all the Baby Bells after they broke up. They didn’t send their VP’s of Marketing they sent the equivalent of their business relations guy. And so you’d go to a meeting and you’d be sitting next to the guy whose title was manager of business relations for Nynex or you pick the company. Well that is not terribly interesting to me to be honest. These are the people that the guys who were working for me had people who interfaced with. So I stopped going. And somebody would call and I’d pick up the phone and they’d say, Lee are you going to the ADAPSO meeting? And I’d say, nope, I can’t do it, I’ve got something else I’m doing. And there were other gatherings that started to take place like Esther Dyson’s conference. Ben Rosen started it, then Esther picked it up and it was a place to go because not only were the CEOs there, but there were CEOs from the surrounding disciplines that we were much more interested in. And you can only go to so many conferences. So I stopped going just about the same time that it switched over to becoming ITAA, maybe a year after. I only went to one or two meetings after that but it was primarily because the people that I wanted to see socially weren’t coming and because the issues that they were addressing were now way down in the heap. They weren’t strategic and a lot of them weren’t even tactical. They were procedural.

Frana: How many people were going the last year or two that you were in ADAPSO? How much had the organization grown at that point?

Keet: I think at the height when I was still there I believe there were six or seven hundred people coming to a meeting but I'm guessing at that. Luanne [Johnson] would give you very accurate feedback on that one. We didn't interact with all the other disciplines as much as we did with ourselves. I mean I knew all the software people. We would socialize and see each other. I got to know a number of the people on the services side and the processing side and so on but it was a big organization. There were lots of people going.

Frana: And lots of committees.

Keet: And lots of committees and the committees became more and more segmented so you could really get into the specialties and if you were really interested in a bill pending on the hill you could find six other people who were going to assemble and it just wasn't of interest to the CEOs of companies at the time. And also my life was changing. As I say, I don't hold onto one thing for too long. I had sold the company to Dun & Bradstreet. I was made president of the software products group in 1979 and I stayed there for four years building the company through lots of acquisitions and trying to build it through some internal intrapreneurship. I was still going to ADAPSO meetings but I wasn't going as often as I had been before. I guess my interest started to decline about then. And then in 1984 I took a year's sabbatical and went to France and that broke a lot of the linkages so when I came back I didn't reestablish them the way they were, and I was in a different business. I came back as an investment banker and venture capitalist and I was on the other side of the table at that point. I did go back to ADAPSO long enough to join a President's Roundtable which is one of the best things that ADAPSO did, but the section that I went into was for those who sold to the industry rather than those in the industry and while those are great people and they are still friends and we're having a dinner tomorrow night, they weren't the guys that I had fought shoulder to shoulder with so it wasn't the same level of interest and loyalty that we had before.

VENTURE CAPITALISM

Frana: And you felt that having your kind of experience before you became a venture capitalist was very important. Were there a lot of venture capitalists that didn't understand the industry?

Keet: I don't want to be too harsh, but the average venture capitalist minted in the 1990s in the great boom of venture capital that started in about 1990 never ran a company, never made a payroll, never fired anybody in their life. Terrific training at great schools and terrifically selected but they looked an awful lot like junior lawyers do in a law firm with no real practical experience. And you can blame an awful lot of the dotcom bust on those people. You would never have found Ben Rosen or Arthur Rock or any of the old time venture guys blowing the bubble up the way it got blown up. It was just too much money chasing too few deals. Actually I should take that back. It was just too much money. And people who didn't have the practical experience to know enough not to just write bigger and bigger checks. And it has hurt the industry tremendously. So, to your question, yes, I felt very privileged to have had an operational background before I started trying to help build small companies. Also the first thing that Vanguard Atlantic did was help big companies and we got a second education when we found out that my experience at Dun & Bradstreet - which was that we'd buy something and kill it - wasn't unusual. The big companies do not do well buying small companies. The internal culture kills them – it just comes in like a macrophage, surrounds them and digests them and that wasn't too soul satisfying.

We made a fair amount of money from the time we founded Vanguard in 1984 until the time we exited the investment banking business in 1989. And we had very good large customers – Coopers & Lybrand, National Semiconductor, Nynex, ADC Communications and others. And in retrospect we didn't do anybody any favors by taking those fees. In general, big companies should not buy small companies was the lesson I learned. But what we also learned was that there was a lot that could be done helping the small companies. Not just to get bought but to build them. And that's why we decided to go for what we called initially investment banking but really was pure venture capital. The difference was we put in talent and not just money but that's a whole different story, and not for today.

Frana: I don't think for today. I did want to close maybe by asking you about your foundation. It supports things like the Trudeau Institute – you have a great love of the Adirondacks that you also support with this foundation. Is this a recent thing that you've gotten into – organizing this foundation to help the Adirondacks?

Keet: Our ancestral home is there. My grandfather was born there and his grandfather was born there. He moved away when he was 16 and he didn't go back and my father went back only at the end of his career. But we bought the family house and that's where we consider home. The family foundation is not exclusively for support of the Adirondacks. The Adirondacks is one of the most beautiful areas on earth but it is also part of the Appalachian chain in the sense of its culture and its people who frequently have been left behind. They are good people but they haven't had the educational or career opportunities. There is no industry. It is so bitter cold in the winter and so remote that the only industry is tourism. If you think it's cold in Minnesota you should visit. I've lived in the Adirondacks in the winter for two consecutive weeks where the highest temperature in the middle of the day was 20 below zero and it went to more than 40 below at night. So we try to help the area retain its uniqueness and its ecology. At the same time we are very interested in helping keep the area not just for trees and deer but for the humans who live there. There are institutions there like Paul Smith's College which educates local kids who want to stay and work and live in the Adirondacks and we try to help them; we help the Trudeau Institute which is a biological research institute that is doing really bleeding edge science at the real fundamental level. Most people don't want to support fundamental science because it doesn't solve the problem of a sister with breast cancer or somebody in the family with Parkinson's disease or something like that, but we do because it will help everyone's grand-kid.

Frana: They do more than tuberculosis these days I'm sure?

Keet: Oh, absolutely. They still do the most advanced lung immunological research which includes of course tuberculosis. And most people think, oh we cured tuberculosis. Well three million people a year die of tuberculosis. It is one of the leading killers still extant that we have not found a cure for. A lot of other things that kill us we have a cure for, but we just don't have a delivery mechanism to get kids the nutrients they need so they don't die from diarrhea. That is a different but important issue. Trudeau is doing really fundamental research. It’s a great place, a little campus sitting on the lake with about 180 people all working in a very collegial environment – some of the best scientists in the world and recognized as such and those are the kinds of institutions we like to support.

Frana: Thanks for sitting for the interview, Lee.

Keet: Thanks for doing it, Phil.