Oral History of Gary Durbin

Interviewed by:
Luanne Johnson

Recorded: May 17, 2000
Oakland, California

CHM Reference number: X5538.2010

Table of Contents

Gary Durbin

Conducted by Luanne Johnson

Abstract: In this interview, Gary Durbin discusses his long career as a technologist and entrepreneur in the software industry. He describes the challenges of being a programmer in the 1960s when it was necessary to compensate for the limited capabilities of computers and the intimate knowledge of the internal workings of the machines that he developed as a result. He describes the factors that led him to found Cybernetic Systems, Tesseract, and Seeker Software and how the difficulty of competing against IBM's announced, but undeveloped, product convinced him that an independent software company providing application software would be less vulnerable to IBM's market domination than a systems software company. He talks about the difficulties that software companies had in obtaining venture capital financing in the 1970s, about the challenges of changing his company from a services business model to a product business model and about the strengths and weaknesses of his competitors in the payroll/HR marketplace.

[Editor's Note: This interview was conducted in Mr. Durbin's home in Oakland, CA.]

Background and Education

Gary Durbin: Writing software in the early days was a matter of people pushing the limits. In fact, the year 2000 date issue goes all the way back to the times when you had to get everything on an 80-column card and had very limited memory in the computer.

Luanne Johnson: Yes, you were limited to 24k for a program..

Durbin: If you had a 24K machine, that was a big machine. The 16k 1401 was a good-sized machine. In fact, the 1401 was 16k max. I remember the addressing schemes of many of the machines were limited to 4k or 8K memories. When we were doing things like check processing, you literally had count the instructions that were going to execute as a check moved down the path, because, if you had too many instructions, when you popped the lever to select a check, you would be too late.

Johnson: Sure.

Durbin: Check sorters had a array of bins (called pockets) – from 12 to 20 bins – that checks destined for different banks or federal reserve banks were grouped into. The check sorters read the MICR characters on the bottom of each check, and then the check was guided along a path over the pockets. The computer would open a lever before a check reached a pocket and the check would be guided into the pocket. The checks passed through the sorter at from 1,000 to 2,000 checks per minute, so the time from the read station to the pocket was very short. Each computer instruction had a different execution time so you had to figure out how to do things with the fastest and fewest instructions. The checks with more complex lookup routines were assigned to the later pockets. The simple ones, the ones that you could figure out right way, you could drop in early pockets. If the compute routine took too long, the check would pass the pocket before the select lever flipped, and the check would miss its pocket and end up in the last pocket – the reject pocket.

We were pushing the limit of those machines and many of us who became systems programmers cut our teeth on those kinds of applications because we had to get very close to the hardware and the operating system in order to get the performance we needed. Those machines were incredibly slow by today's standards. I think a typical memory cycle on a 1401 was 11 milliseconds - not microseconds, milliseconds.

So we learned the craft, we learned optimization, we learned how the machines really worked. And that was the hallmark of software in the 1960s. Because the machines were so limited, the software ended pushing the hardware. The machines were never fast enough for what we wanted to do. We were always spending a tremendous amount of time dealing with the shortcomings of the machines - the storage, the disks, the tapes, and so forth. And the hotshots were those who figured out things like how to optimize tape on data channels, or use disk for overlays, or multi-thread telecommunications.

Johnson: And that's when everybody's got this big shock about how expensive it was going to be to program these machines because there was so much time spent doing things to get the machine working the way you really needed to have it work.

Durbin: Yes. A lot of time was spent in optimization. When the higher level languages like Cobol came along, they didn't operate as efficiently, but they saved time in programming. But that created more demand for faster and faster machines.

Now I think it's changed. The software doesn't push. The software takes advantage. We're now able to do things because faster machines are inexpensively available.

Johnson: Sure. Let's go to my list of questions. Give me some of your personal background and tell me what it was that positioned you to go off on your own.

Durbin: I was from a fairly poor background; my father was raised on a ranch. So I think that that ranch mentality was something he taught me. When I grew up if something went wrong, like a water pipe or something, - we fixed it. I didn't even know that you could call a plumber until I was living in an apartment.

Johnson: *(Laughter)*

Durbin: Seriously, I was probably in my 20s before I actually heard about someone calling a plumber. What a novel idea. So I had this can-do kind of attitude. Which is what an entrepreneur has.

Johnson: Was that in California?

Durbin: I was born in Santa Monica and spent most of my early years in Oakland. My parents were from Idaho and were dislocated by the war. The moved to Santa Monica so that my father could work in the aircraft industry.

Johnson: But that attitude was inbred in him, and he transferred it in you.

Durbin: Yes. So in thinking back as to why I'm an entrepreneur, it has a lot to do with that attitude you develop living on a ranch where you do whatever you have to do to make it work.

I went to UC Berkeley majoring in math and physics, but after my first year I had to drop out to make some money. Because my family couldn't support me through school. I started college when I was 17, so here I was at 18 trying to figure out how I was going to manage to get a job that would pay enough so that I could make my way through several years of college.

I thought data processing sounded interesting. I had some knowledge about it because of the computer lab in the math department. I hadn't yet had any experience with it, but I was curious about it. So I borrowed some money and went to one of the data processing schools for tab equipment. The course seemed expensive at the time, but I expected it would pay off

Johnson: Which school did you go to?

Durbin: Automation Institute. Going to class was like mind candy; it was so much fun and this material was so easy. I couldn't believe it. Partly it was because I had a background in electronics.

Johnson: Oh?

Durbin: As a kid, I experimented in electronics and made spare change in high school fixing radios and televisions in our neighborhood. I built an oscilloscope, signal generators, and that kind of thing. I would go out to an Army Surplus store by the airport and pick up electronic parts. You could buy radar tubes for a buck. I would wander around the store and find interesting parts or assemblies and then build those into experiments.

First Programming Job at Wells Fargo

So the electronics and my background in mathematics made this stuff a piece of cake. Because it was so easy for me, the teachers thought I had a natural talent for it – I think I probably do. When I was getting close to graduation, one teacher who worked for Wells Fargo said, "Hey, why don't you come down to Wells Fargo and apply for a job. We're looking for programmers."

So I did. And they gave me aptitude test. Remember programming aptitude tests?

Johnson: Oh, yes.

Durbin: This test was so much fun. They gave you an hour and 45 minutes to take the test. I thought that something was wrong because I finished it so early. The test was mainly little exercises in symbolic logic. I used to work symbolic logic problems to pass the time on the bus to Berkeley. So they gave me a job right away, and put me in a programmer-training course.

The machine I first learned to program was an IBM 650. It was a tube machine with a 2000-word - 20K decimal digits - drum memory - slow beyond belief. This machine forced you to optimize or die.

It was so slow because it pulled the instructions off the drum. You had to compute where the location of the next instruction was going to be. While an instruction was executing, the drum would be spinning. You wanted the next instruction to be right under a read head when it was time to fetch it..

The compiler for the 650 was called SOAP, Symbolic Optimal Assembly Program. We were on the second version of it. I remember that it was SOAP IIa. And compiling was a big thing, because the system I was working on was real estate loans and the main program for it was about 3,000 statements. It took most of a day to do a compilation. You'd feed it in and the compiler would munch and chomp. The probability that you could get the whole program to feed in without a card jam was slim.

So you'd compile every once in a while, but in the meantime you'd do patches. We'd write instructions in machine language and patch them in. And every once in a while, every couple of

weeks, you'd have to go through the process of putting all the patches back into the original source, compiling it, and seeing that you got the same results after you compiled it - tedious.

Working at Wells Fargo was a great experience for me because it was so much fun. And that was in the days when programmers were not professionals, so they had to pay us overtime. So they would pay me a salary, which was double or triple what I'd been making before, and they would come around and give me these overtime checks. There were times when I had overtime checks for several weeks just stuffed in my coat. I didn't know what to do with them all. It was just amazing.

And then every once in a while, they would do this great thing and send you off to school to learn a new programming language like Autocoder for the 1401. Then the IBM 360 came along. Wells Fargo had one of the first 360s. By then I was one of the people they gave the difficult problems to. One day my boss walked in with a couple boxes of cards – a couple of thousand cards – and said, "These boxes are supposed to be an operating system for the 360. Don't ask me where I got them. Your job is to figure out what the hell it is." The first thing I had to do was to interpret the cards so that I could see what language it was written in.

It was an operating system that had a basis in DOS, which was the first operating system for the 360. But it had major modifications to support check readers. My job was to figure out how these modifications worked and then come up with prototype programs that would actually use those modifications and drive the check sorters from the 360.

The first thing I did was to go over to the IBM support center in San Francisco and say, "What kind of compilers do you have here?" And then figure out which compilers could handle this stuff, and then get it compiled and figure out what it was. It was great fun.

My boss at Wells Fargo was Bernie Hogan. He's someone you should talk to.

Johnson: Was he the founder of Hogan Systems?

Durbin: Well, before that he founded Corporation S which was the first OCR scanning company. Then he was at UCC which bought the second company that Bernie started. And then he started Hogan Systems. So he started three software companies.

So my boss at Wells Fargo was Bernie Hogan. He and I cut a deal because I wanted to go back to school, but he had a project that he wanted me to finish, and I was getting all kinds of job offers. The deal was that I would stick around until the project was over, and I could work part time while I was going to school, full time during the summers as long as I wanted. What a deal.

Johnson: Yes, right.

Durbin: And it was at a good rate. Twelve bucks an hour or something like that. Some mind-boggling number in the early 1960s. So I finished the project, and I went back to school.

My plan when I went back to school was to get a degree in computer science. But when I went through the course catalog, there was a problem. By the time I went back to school, I could teach all the classes. I'd been in the industry about three years, and I'd been working on leading-edge telecommunications and operating system. I worked on the first voice answerback system. IBM made a machine called the 7770 that had a 64-word vocabulary. We built a teller terminal system with voice. A customer would come in and the teller would pick up the phone, type in the account number, and it would say: the current balance is - whatever it was. The tellers didn't have terminals in those days. Now tellers all have terminals. But they didn't have those then.

Part of the challenge was to get all of the communication into 64 words. Then once we spec'd the vocabulary, we sent it off to IBM and they programmed the vocabulary. You had your choice of male or female voices; there were three or four of each. And then we programmed the system. It was a challenge because of the limited vocabulary.

So that's the kind of stuff I had been working on when I went back to school. What they were teaching in computer science class was programming and system design. I was like, "Oh, give me a break." That was a big disappointment for me, because I really wanted to get a degree. But ultimately, I ended up not getting a degree because there was no point. I certainly didn't need a degree to get a job. And of course now we were into the Vietnam Era so I stayed in school as long as I could.

Johnson: *(Laughter)* Did you go to PIE [Pacific Intermountain Express] from Wells Fargo?

Durbin: Yes, that's right.

Johnson: Is that when you decided to go off and start your own company?

Durbin: Let's see, how did that work? I went back to school, and then I dropped out again and went back to work for Wells Fargo full time doing primarily telecommunications stuff. And that was how PIE got a hold of me because they were looking for a telecommunications and operating systems expert.

That's when I discovered consulting.

Founding of Cybernetic Systems

Johnson: You can make a lot of money consulting.

Durbin: I went off with four other guys and started a consulting company. It was called Cybernetic Systems. It was headquartered in Berkeley. We had a little office in downtown Berkeley, and we did some fun stuff. We built one of the first online claims processing systems for dental claims. It was an online, real-time, system. We started with a CRT display system, the IBM 2260, which had small character screen, 480 characters – 24 characters wide by 20 lines. We had to build the whole application in this itty-bitty format.

Then when IBM unbundled services in 1970, we got into training.

Johnson: Training, OK.

Durbin: That started, literally, January 1970. Most IBM instructors had taken a class in how to teach a class, but they didn't have extensive field experience. Three of us who did have field experience put together a series of classes that provided in-depth training on IBM's operating system. Our student reviews were excellent because we could explore any topic. But there was only so much leverage to the training. And I got to the place where, after I'd given the same class a dozen times, it didn't have the same edge. The first several times you gave a class after you designed and wrote it was great fun. I really enjoyed that.

But it became repetitive. I finally decided that something had to change when I realized that even though the students were awake, I was asleep. Normally, it's the other way around.

Johnson: *(Laughter)*

Switch from Consulting to Software Products

Durbin: I got interested in software products. The first product we did was called SMR, System Management Reporting System, and that was an SMF analyzer. That was 1971.

Johnson: What does SMF stand for?

Durbin: SMF was the data that the IBM operating system wrote to a file abou the resources being used – CPU time, EXCPs (input output activity), memory and so forth. This was a tool that you could use to analyze what was going on inside the operating system. It also did job accounting. This way companies could allocate the computer time to jobs. This data was used to invoice computer users, to charge back costs internally, or to allocate computer resources. These were pretty expensive machines, so everybody wanted to make sure that they could allocate costs for their usage.

That was an interesting experience because here we were a couple of technicians who wrote this product, and we had no idea of sales or marketing. We had this idea that all you had to do

was to list it in [Larry] Welke's catalog [The ICP Quarterly] and people would beat down the doors.

Well, it wasn't quite like that. It might have worked for payroll systems where there was definite demand, but for job accounting systems, most people could live without them. And we went way overboard. We analyzed everything. We had reports that came out of the product that took every parameter of the operating system that was available and analyzed it. We had charts and graphs that took data that most people didn't understand and showed it several ways.

SMR was way over-engineered and under-marketed. We had no sales force and no marketing. We sold maybe a dozen copies of it, mainly to companies around the Bay Area where somebody would hear about us and get interested. By word of mouth it got out, but it wasn't a raving success. It was good software, but it wasn't a raving success from the product point of view.

So we continued to do classes, and began doing more and more technical consulting. Our particular expertise was custom database systems and custom telecommunications systems. We had a reputation as IBM operating system wonks. And that was how we came into the next opportunity.

I got a call from a friend of mine at one of the rent-a-cop companies. They had a contract with Crocker Bank to build a software security system because Crocker was worried about the security of their computers. They had figured out that physical security wasn't enough, that they needed some internal system protection. They didn't believe, and rightly so, that the IBM's operating system had enough security. This was in 1974.

They had started up this project and brought in some guys to do computer security. But the project was not going well. It had been going on for about nine months, and there weren't any deliverables. So they asked us if we'd come in and take a look at it.

So we did and said, "These guys don't have a clue." They don't know anything about the main thing they're trying to work on, which is the IBM operating system.

It turned out there was a system programmer at Crocker who had built this little operating system modification that did some trapping of faults, catching people who were trying to get into the wrong things. He'd done it because, as a systems programmer, he was trying to protect his stuff.

We looked at that and said, "You know, this guy's got a really good idea. I think that his idea could be expanded and commercialized. We will do that if you will be the first customer. We'll take it and commercialize it, if we can come back and install it here. You guys can be the alpha

client, and you get the results for free." And so we cut that deal.

That product became Secure, which was a product marketed by Boole and Babbage later on. We built it in 1974. By this time we had figured out that sales and marketing needed to be part of the picture. So after we built the product and had a couple of beta clients, we actually hired a salesperson to go out and sell it. He was an ex-FBI fellow who was about six-foot-six. He looked the part. He started selling it, and things started happening.

Competition from IBM

So in 1978, we had maybe a dozen clients. And we went to IBM and we said, "Hey, IBM, wouldn't you like to market this thing? It's the first software security system for IBM machines. We're hearing people say that they think there's a need here." We proposed this to IBM and they came back and said that they were going to do something competitive.

That sort of scared us a bit. And sure enough, about six months later they announced RACF [Resource Access Control Facility]. Of course there was no product yet. There was just an announcement.

Johnson: Typical.

Durbin: And that announcement killed our sales. I mean sales went to zero for six months. This sales guy virtually starved to death. We were advancing him commissions to keep him going and wondering if this product was going to survive. Fortunately, it wasn't our bread and butter - we had other business - but we had invested a lot in the security product.

Johnson: Right. There are so many stories like this. There was so much focus on unbundling, but the real impact was IBM's whole way of operating. Unbundling was almost incidental. It was this kind of stuff that really created problems for the software vendors.

Durbin: They were just as predatory as Microsoft is today. It was terrible. We seriously considered just blowing it off. There was six months of no sales. With no marketing and our ex-FBI guy, we had an installed base of about a dozen customers. Not big time, but we could see that there was a need for a security system. That's what McAfee and those guys are all about today.

We could see the need and auditors were beginning to question whether or not these machines were secure. And they weren't. It was easy to break into them. You could just walk right through them. And the kinds of controls our system brought to computer operations was just amazing.

At Crocker we caught an IBM SE who went in and tried to put some patches into the operating system in the card reader. Because he had access to the computer room, he just went in and did it. Well, this is the bank's online system. And if that patch had not worked, he'd have crashed the whole thing and they wouldn't have had backup. Because the way you should apply patches to make a backup. You did it during off hours and you brought the system up in controlled conditions so that you could fall back right away.

But this guy was sure this patch wasn't going to hurt anything so he just stuffed it in the card reader. And our system stopped him because he was trying to modify system libraries. And, literally, bells went off because there was a console out in the security guard area and, if one of these faults occurred, it rattled out. And this security guy is sitting there with this console that has never had a peep and all of a sudden it fires off a notice. He's got his hand on his gun trying to figure out what's going on.

Johnson: *(Laughter)*

Durbin: *(Laughter)* So that was how it was. In the early days, there was no software security. It was all physical security, but once you were inside the data center, everything was wide open. When IBM finally put out the specifications for RACF, we had something to compete against. What they had done was what we couldn't do on any scale - create awareness in the marketplace. Our product was vastly better than theirs and sales took off.

Johnson: Aha.

Durbin: We went from going along, trying to create a message, trying to tell people about the fact that almost anybody – this was in the early days of TSO (Time-Sharing Option – one of the first systems that allowed people interactive access) – could log onto TSO and go "Scratch Operating System" and – poof! The next time operations tried to boot the operating system - it would be gone.

Johnson: *(Laughter)*

Durbin: That's how exposed things when we were entering into the era of time-sharing. There was just one hole after another.

Sales took off and we were going gangbusters. Then one day I got a call from Bruce Coleman [President of Boole & Babbage] saying, "You know, we think this is a hot area and you guys have got the only product besides IBM." We were there first, and it was a better product. So whenever we'd compete head-to-head against RACF if people didn't think they had to buy the IBM system because it was from IBM, we'd win the sale.

IBM was always going to have a big piece of the market share simply because, without thinking, people would buy the IBM product just because it was from IBM.

Johnson: Sure.

Durbin: But it was different when we got technicians involved, system programmers who could see the differences between the two products. The big difference in our system was that it was vastly easier to administer. What was happening was that the system programmers would buy RACF, but they wouldn't implement it because it was just too tedious. Our system was much easier to implement. So we had a much better record of customers actually using it.

Bruce came along and said, "We're interested in buying this." I sat down and worked out the numbers. At this point Boole had a worldwide sales force of 25-30 people, something like that. Vastly more than what we had. With their sales force we would make more from their royalties than we could going it alone. So we cut the deal. And they went gangbusters. This product was the fastest moving product Boole had ever seen. I think in the first two years, they sold 1200 copies.

Johnson: Wow.

Durbin: It was really hot. And they were selling it at four times what we were selling it for. We were selling it at $10,000; they were selling it at $40,000.

Johnson: Did the product get an ICP award?

Durbin: Yes, it did. We hadn't sold a million dollars worth, but Secure certainly did. And it was one of the strongest products at that time. Its success was in some ways bad for the product. Since it was so hot, Boole didn't invest in the product and take it to the next stage.

That happens a lot when you have an entrepreneurial product and the entrepreneurial engineers are no longer around. We did class after class after class, we did all of the knowledge transfer, and those guys could maintain the product, but they couldn't build it, they couldn't develop it.

Johnson: They couldn't take it to that next level.

Durbin: They didn't have a vision. They understood the software, but they didn't have a vision. Once you lose the vision guys, software tends to not move at the same rate. So Boole was very successful with sales and marketing. As part the deal, I signed a non-compete agreement that included a prohibition on activities in software security for seven years. Since I was the author of the product, they wanted me out of the business.

Johnson: Sure.

Durbin: We got a lot of money for that agreement so it was an okay thing. But I had really been burnt by trying to dance around of the feet of the elephant. If IBM came into your sphere, you were toast.

Johnson: Right.

Durbin: I remember that in the mid-1970s there were about four or five transaction processing monitors besides CICS. There was Intercomm, there was Shadow, there was Task/Master - and Cullinane had one. And four years later they were all toast. They were all just gone. We had built a transaction monitor that our client was running beside CICS and it ran rings around CICS. But we abandoned the idea of trying to market it. There was no way we could go up against IBM on their turf.

Johnson: Lee Keet [founder of Turnkey Systems, developer of Task/Master] is one of the people I interviewed, and he said their only choice was to focus on getting their other products going to take up the slack.

Durbin: IBM had decided they were going to own the transaction processing marketplace, and they just killed everybody else. You couldn't compete. If IBM decided they were going to own it on their hardware, you couldn't compete. They would price you out. They would market you out. They would leverage you out. You were gone.

I had a taste of that with Secure, but IBM had not made security a strategic space. They had come out with a product because their customers wanted one, but they hadn't decided they had to own the space. That was the only thing that saved us – they let us compete.

Johnson: And you can see why they would decide that.

Durbin: That's right. With database, they decided the same thing.

Johnson: Lee talks about getting the IBM representative to ADAPSO to arrange a meeting for him with the top-level IBM people so he could explain to them what they were doing to his market. He said he still respects all of those guys, they dealt with him very professionally, very graciously, but they didn't give an inch.

Durbin: Right.

Johnson: All he could do was to decide to focus on other products and get out of that market.

Durbin: Yes. So I was smarting from that. That six-month dry period, boy, that was a bad time. That was a really bad time. I had been thinking I was going to see our whole investment in this product go down the tubes. For the first time, with that product we had hired a guy full-time to work on that product. We were a little consulting company. There were a dozen people in the company. So putting somebody full-time on that product meant that all the rest of us were working to support that effort.

Plus other people in the company were still working on it. I was the author, I was putting in a lot of time, and my partner, Pete Lamasney, was putting in a lot of time. So we were working on the product in addition to going out and hustling gigs to keep the company going.

Shift to a Focus on Application Software

It was a big investment for a little firm. So I was really smarting about that. It was in October 1978 when we did the deal with Boole. I said, "Look, the thing that IBM is not going to do is go into the application space. They don't need to own that space." We had previously built a human resource system for Bechtel – an online, real-time HR system. At about the same time we were building the security product, we were working on a custom contract for Bechtel.

We did it in such a way that we ended up owning the rights to the product. They couldn't really pay for a complete product, so we got it in the backdoor by charging them for services. We built the product on spec and then charged them for the usage of it.

Johnson: That's how most of the software industry was funded originally, wasn't it?

Durbin: You got it.

Johnson: Everybody did it that way.

Durbin: So did we. And what we built was vastly different from what InSci had done. InSci was the market leader at that time. But we had built a system that was a completely online, real-time system built on a database model. I had read Dr. Codd's papers on relational database when they were first published, which was in the mid 1970s. They were originally AFIPS [American Federation of Information Processing Societies] papers. Remember AFIPS?

Johnson: Yes, sure.

Durbin: So some of them were published by AFIPS and some of them were published by the IEEE [Institute of Electrical and Electronics Engineers]. I gathered them all up. I got interested in relational database ideas when a friend of mine who was at Berkeley doing graduate work said, "You know, we got this funny thing over here called System R. And you

ought to take a look at this, this is a little different."

I met him on campus where he had a timeshare terminal into the computer center. He did some queries I said, "Wow, this is pretty cool. Where did this come from?" Then I did some research, and I found Dr. Codd's papers. I said, "Oh, this is music to my ears." Somebody had thought through all of issues with manipulation of data, the relational calculus, the relational algebra, - how manipulating data could have a mathematical basis. My background was math so these ideas made perfect sense to me. When we started building the Bechtel system, we built a little relational database inside the system. The whole architecture of our early products was based on a relational model. But, of course, we didn't have a relational database to work with. So the initial systems were BDAM and VSAM – low level data access schemes, and we had to build a lot of infrastructure on our own.

So we built the system for Bechtel and they used it, and that was pretty cool. In fact, right after they went live, I thought, "There's a product here."

Johnson: Now this was done when you had your consulting firm?

Durbin: Yes. This was at the same time we were working on the security product. We finished the product for Bechtel in 1977 and we went out to try to market it. We said, "Here it is, a real-time, online HR system." And people were saying, "Are you out of your mind? We don't have our receivables online, we don't have our payables or our accounting online, why would we want HR online?"

Bechtel was a leader at the time, and their business was all about people. So they recognized a need. And there was a guy there that had a vision, and he wanted to have all kinds of access to this information. It paid for itself in no time. But we said, "There's no market for an online, real-time HR system." So we put the HR product back on the shelf, and went off and did other things.

Then fast-forward two years later, we'd sold Secure, we had some money to make reinvestments, and started looking around for the next thing to do. I'd been burned really badly by being in the system software business, so I was thinking again about the application software business. And IBM announced Interpers, Interactive Personnel System, a product that they were going to sell.

Johnson: Okay.

Durbin: One thing that did happen with Secure, which we called DAS, Data Access Security system – we marketed it as DAS and Boole marketed it as Secure – was a coattail

effect. IBM created consciousness for security, and then the buyers were more educated and we could compete. We could sell into a developing market.

So I said, “Is there going to be a coattail effect?” The fact that IBM announced an interactive online personnel system means that somebody thinks there’s a market for one. InSci didn’t have an online system, though they were talking about doing it. And here we already had it. It had been in production for several years.

So I said, “This could work.” We took the royalties that were coming in from Boole and reinvested them into the HR product. We completely refurbished in internally and generalized it for the personnel/HR market.

Challenges in Changing the Business Model

We had an interesting thing. We sold the security product to Boole in October of 1978. About the middle of 1979, we decided that the application business was where we were going. We had an off-site meeting with the executives of the company. We said, “Okay, we’re going to become an application product company. What does it take to do that?”

I think we had three days off-site. It was one of those meeting with all of the papers up on the wall, action lists, subject breakdowns, and so forth. We had identified the things we needed to do to change a company that had its feet in two areas – systems software on the one hand and consulting on the other – to become an application product company. So we came up with this strategic plan with all of these tactical action items.

Johnson: OK.

Durbin: It was a well-thought-out plan as it turned out, and we executed on it. I had a chance about a year later to look back at how we did and we did really well. But a consequence of executing on the plan was that I had completely swapped out the management team. I looked back and realized that, even though the executives had signed up for it, going from a consulting company to a product company was such a huge dislocation and such a radical change in thinking that most of them didn’t make it.

I brought on some other people and those became the stars. And the people who had been the stars earlier were gone. I had a woman VP who was fabulous. And at one point she came in and said, “I understand the consulting business but this product business - I just don’t get it. I don’t understand how to do it.”

And I had a partner, a major shareholder of the company, who had worked out a spreadsheet and said, “You can’t make money in the product business. Every time you sell one, you’re increasing your cost and there’s no way that you get the economies of scale that you’re talking

about. It can't happen." He was convinced. He had a model and that model just kept getting deeper and deeper and deeper into red ink.

I said, "Well, I have another model; I don't think that's the way it works". So we argued over whether or not it was going to work, and he went along with the program for awhile. But about a year later, he said, "Buy me out. I want out of here." So for about $100,000, he sold his 40% interest in Tesseract. Three years and two rounds of VC capital later, the company sold for $12 million. But it wouldn't have made any difference. Because he had this burning feeling in his stomach that he couldn't handle.

So taking that company through that transition from a consulting and system software company to an application product company was gut wrenching.

Johnson: That's fascinating. I don't know of many companies that have successfully completely changed their business model. Comshare is one other that I'm aware of because they went from a time-sharing company to a software products company. But I know that it's extremely difficult. As you said, you had to swap out your entire management team in the process.

This is one of the reasons why I think it's so important to preserve the business history, not just the history of the technology. It's not enough to have great technology like your online, real-time HR system. If you can't create a company to bring it successfully to market, it's meaningless.

Durbin: Sure. I have to say, though, that at Tesseract I never figured out the marketing side of things. We did okay. But we should *never* have lost a sale to Integral. We had vastly better products. But they had these suede-shoed guys from IBM who really understood marketing. They could out market us and out sell us.

We held our own. And we ultimately ended up with a bigger company than Integral, and our product killed InSci. InSci tried to buy Tesseract.

Johnson: Oh, I didn't know that. Who was running it at that point?

Durbin: It was the guy from Welsh Carson Anderson, Bruce Anderson. He was the guy that got in touch with me. We were beginning to be in a pitched battle with them because they were the market leader. We were coming in as the new kid on the block, and we were threatening them. Any time a customer recognized that the future of the application was online, we were winning the business. We were beginning to really hurt their sales.

They came to us and said they were interested in an acquisition. So I flew to New Jersey and spent some time with them. Anderson was from ADP, so he understood the payroll side of the

business. At that time, Tesseract did not have a payroll system. We were the first guys to come into this business from the HR side.

We didn't have a payroll system, so an advantage would have been that we could have gotten a payroll system. The disadvantage was that we were hearing lots and lots problems with their products – customers were unhappy. They had gone through several management changes and then they did what so many companies have done – the thing that Sybase did. They came out with a bad release, a release that was full of problems.

The customer base can turn off quickly under those circumstances. If you're Microsoft and there's no alternative, then they sort of tough it through. But in the case of InSci, there was another alternative - us.

Not too long after that, Integral was coming on strong. So both of us were stepping on their heels. They said they were serous about acquiring us and started discussing terms. But I believed that we could beat them. I didn't have to let them buy me. I was working on raising money, and I had a couple of alternatives. I managed to get VC's to the table. So we went the VC route, because I decided if I had the capital, I could beat InSci.

Challenges in Finding Financing

I had an interesting experience with the VCs in the early days.

Johnson: Before you talk about that, when did you actual change the name of the company to Tesseract?

Durbin: When we became an HR product company is when we changed the name about 1979.

Johnson: Okay. So what's your story about VCs?

Durbin: When I first went out to raise money from VCs, I can remember the sound in my ears of VCs saying, "We don't invest in software." I'm sure you've heard this before.

Johnson: Oh, yes. I've had people say things like: If you were a software product company owner and a VC saw you coming down the street, they'd cross to the other side. They wanted nothing to do with you.

Durbin: Nothing to do with you. I went down to Sand Hill Road and I talked to VC after VC. And so many of them said, "We don't do software." Well, then I finally found Greylock, who had invested in Cullinet. And they had been working with Sutter Hill. And Sutter Hill had begun

to realize that there was something happening here and that software companies might make sense.

So Sutter Hill and Whitney came in as my first investors, and then Greylock. But it took me probably a year and a half to find VCs.

I had difficultly financing Tesseract. In the early stage, there was no venture capital available, it just wasn't there. So I did a couple of rounds of sort of creative financing.

Johnson: Such as?

Durbin: Well, one round we got from Online Business Systems which Jeff Stein was running. Because this was mainframe software, we had to buy computer time, and we were buying it from Online Business Systems. I had to negotiate a really long payment schedule, like 90 – 120 days. And then I realized that if I looked at my balance sheet, the biggest single number was this big payable I owed to Jeff. And I thought if we could just capitalize that debt, we'd be in great shape.

So I went to Jeff and said, "Jeff, this is the situation. I owe you a lot of money. I want to turn that into equity. And while we're at it" – I think I owed him about $90,000 – "how about another $90,000, let's make it $180,000."

And he said, "So what's the deal?" I said, "Well, the deal is that if you put this in, I'm going to continue to finance the company and so I'd like a buy back. I'll double your money. And if I can double your money in 18 months, cut me that deal." And he said okay because the downside was he'd own a big piece of the company; upside he doubles his money.

Johnson: Okay.

Durbin: I had been unable to find any VCs so I just pulled back on looking for VCs and I did this feathers and shells deal with Jeff. And then about a year later, we went back to the VCs and they had started to think about software. I found Sutter Hill. And it was really great that Jeff had made the deal he did because I was able to take a piece of the money I got from the VCs and buy him out, so that the VC deal was almost anti-diluted.

Johnson: Oh, okay.

Durbin: So take him out, bring the VCs in, and still hold on to a big piece.

Johnson: How cool.

Durbin: The thing that I didn't understand was how much capital a software product company can burn through. If you want to focus a business on products, and you have an emerging product, you've got to have a lot of capital.

Johnson: That's one of the big differences between products and services, and that's why people have a problem making that transition, because they just don't anticipate what it's going to take to get that product out there.

Durbin: I thought I understood. I knew that the product was going to take more capital. We'd done this deal with Boole, so we had some capital from that to reinvest, and then I raised a little bit more with a private financing through some guys I knew. And it still wasn't enough, so I went to Jeff and raised some more. And then it became really clear what it was going to take, so I started raising VC money.

Sale of Tesseract

Johnson: So how long did was Tesseract in existence before Ceridian bought it?

Durbin: Well, actually, I sold it twice. First time I sold it, it was to Prudential.

Johnson: Then you bought it back from Prudential, is that what happened?

Durbin: Well, I put together an investor group to buy it back.

Johnson: Okay. What was the timing of that?

Durbin: August of 1986 is when we sold it to Prudential. We had been trying to raise another round of VC money and our business plan had gone to Prudential Ventures.

Johnson: Oh, okay.

Durbin: Prudential Ventures knew that Prudential was looking at some strategic investments they wanted to do. So the business plan ended up going over there. And they had a guy there named Les Ralson, who was truly a visionary in an insurance company. He was amazing. The guy had been a fighter pilot in Korea. He was fearless.

His idea was to bundle payroll service and insurance, medical insurance, to provide a more comprehensive service and thereby, to lock in his insurance business. They'd done the numbers. They could give the payroll processing away for free if it would just add a year to the longevity of their medical insurance.

So it was a no-brainer. And we had the payroll system to do it. There was only one thing wrong with this plan. And the one thing wrong with this plan was that insurance guys can't sell payroll systems. I mean the ADP sales force and the insurance sales force are like night and day. So they just couldn't handle it.

Johnson: When did you add payroll? When did that happen?

Durbin: Well, that was one of the reasons for getting financing.

Johnson: And did you develop that in-house then?

Durbin: The guys who had developed the payroll system for InSci had left InSci and built another payroll system with a better engine. They had sold two or three copies of it. So we bought that system from them and dramatically re-engineered it because it really wasn't designed around a database. From them we got the basic computational engine and we picked up a bunch of expertise.

So that was about a $2 million project to acquire that and re-engineer it and roll it out, which in 1984 was a bundle. That's why I knew I couldn't afford to bootstrap and that's why we had to raise money to do it.

Johnson: Okay. So you sold Tesseract to Prudential in 1986, so when did you buy it back?

Durbin: I think it was 1991. Prudential was a great thing for me because right after the Prudential deal, they asked me to come take a look at the claims system that they had. Tesseract had a claims system as well.

I looked at it. And, to make a long story short, ended up making a proposal to Prudential that I start up an R&D group.

Johnson: Okay.

Durbin: They thought that was a good thing because there really was no intense R&D. So we looked at the whole emerging artificial intelligence technology. In 1987 there was a lot of interest in AI.

Now IBM was promoting AI because they thought this thing would burn up lots and lots of clicks. And of course, it did. But it just didn't do much.

Johnson: *(Laughter)*

Durbin: *(Laughter)* The machines were getting faster and they thought that the software wasn't going to keep up. So they had the idea that the relational database, DB2, and AI were going to burn up all these excess clicks.

But there was some real technology there. There were some really good ideas. We ended up building a prototype for a massively parallel inference engine. We patented the prototype, but it was at least ten years ahead of the market. The peer-to-peer applications that are part of the the Internet, that's what this software was all about. Napster and Gnutella and so forth. Well, the ideas in those are the ideas that we were working with back in 1988.

And there's a technology development center that Siemens put together in Berkeley and they've got some agent software. I'd been talking to them and I referred them back to my patent.

Johnson: Do you own the patent?

Durbin: Tesseract does.

So I was out playing in the sandbox, having a good time. Meanwhile, Dave Duffield [Founder of PeopleSoft] was off creating client/server stuff. Tesseract was the leading HR software up until the client/server stuff came out. So I'm out playing in the sandbox but I attended the executive sessions at Tesseract and clearly we were getting heavily hit by this new guy, Duffield.

And so I put together a plan to build a client-server product and went back to Prudential and said, "Okay guys, it's time to step up and spend some money." Well, these are insurance guys. They said, "So we spend the money and *then* you find out if it's going to work?" These are the guys that, you know, you pay them and then maybe they pay your claim, right? They get the cash *up front*.

Johnson: Sure. You give them cash for years before they ever give you any back.

Durbin: That's right. Their whole idea is we get your money, we hold on to it, and then maybe we give some back. So their response was, "You mean we invest in it, but it might not sell?"

"Well, yes. High risk, high rewards. That's the way this works."

And if you said the word "risk" the walls would shake.

Johnson: *(Laughter)*

Durbin: These are the most risk-adverse executives on the planet and they should be. They're an insurance company. The CEO of Tesseract at the time was Bill Luckenby. And we beat our heads against that wall for a year.

And meanwhile Duffield has taken off and I'm getting frustrated. I'm chewing my arm off because it's clear that we have to do something. So I went to Bill one day and I said, "Bill, why don't we talk to the board and say that we want to go see if there isn't a way to fund this outside, because these guys are never going to do this. We're just wasting our breath."

We talked to the board and they said okay So I went out and found a VC, Bob Feibush. And we put together a package and bought the company back.

Johnson: When was that?

Durbin: That was 1991.

Johnson: Okay.

Durbin: We were in the process of dramatically changing the direction again. We had the premier product, and we were going to fight Duffield. We bought the company back for about $20 million. And about a year later these guys from Ceridian showed up and they've got to have a new payroll solution. They were running a payroll system that was built back in the SBC [Service Bureau Corporation] era. Written in DOS Assembler.

And of course we've got top end DB2, the absolute latest state of the art. They fall in love with the system, they want to buy the company. Eighteen months later they bought it for $55 million.

But it turned out that Ceridian really couldn't absorb it. The technology gap was just too big. To go from sequential disk – I mean this is 1993 and they were running a sequential assembler disk system. At least they'd converted it from tape to run on disk. And they had 30 production centers where they ran this DOS stuff.

Johnson: How could they even find people who still knew enough about DOS assembler to even keep the thing going?

Founding of Seeker Software

Durbin: Well, that was a problem. So here we come in with this relational, MVS, state of the art system. And they couldn't get there. They just couldn't reach it. And ultimately the project was not a success and they wrote off $150 million. But I was gone before that because it was clear to me what was happening. I tried to be involved in the project but it just wasn't

working out. So it was time for me to go do something else. That's when I started Seeker Software.

Johnson: So what was Seeker?

Durbin: Seeker was Web Business Applications. Primarily what's called B to E, business to employee. It started off with simple things like the employee directory and people thought that was pretty cool. But the thing that really hit was when we took on the task of automating manager transactions.

If you're a manager and you want to hire somebody, fire somebody, move them around inside the company, and so forth, it's astounding how much time it takes.

Johnson: Oh, sure.

Durbin: So we could go on the web and automate the *process*. This was not about automating the transaction, but about automating the process. So you would say, "Well, I want to terminate somebody." Well, what kind of termination are we talking about here? Voluntary, when they walk in and quit, or are you going to fire them? OK, if you're going to fire them, we've got some work to do because you've got to paper the walls, have this thing reviewed, get these approvals, etc. etc. It helps the managers do all that stuff. Steps them through it click by click by click.

Johnson: Who bought that from you?

Durbin: Concur Technologies.

Johnson: Are they expanding it?

Durbin: They didn't really understand the vision. Again, there was a technology gap. They do automation of expenses and that's a process as well. You type in your expenses or they had really good interfaces with the credit card companies. So you'd go out and charge stuff and by the time you got back to your office, that stuff would already be entered in and you go click, click, click, yes it's okay. And you get an expense check.

So they were the leader in that space. We looked at expense as one of the areas we were going to end up automating. So from the strategic point of view, it made all the sense in the world, but the corporate culture just didn't work. From a product and customer point of view, it should have been a marriage made in heaven. But it's the other part of the business that has fallen on hard times. The HR division has actually been on target.

Johnson: So now you're doing angel investing?

Durbin: And advising.

Johnson: Are you're sitting on some boards?

Durbin: Yes.

Johnson: Does it keep you occupied enough?

Durbin: Probably. I don't think I want to be a CEO again.

Johnson: Well, thanks for your time, Gary. It's been great talking to you. You have a lot of wonderful stories about the early days of the software industry.

Durbin: Thank you. It's been fun.

