AI: Expert Systems Pioneer Meeting
Session 4: Business Reviews of Initial Companies

Moderators:
Burt Grad
David C. Brock

Recorded May 14, 2018
Mountain View, CA

CHM Reference number: X8652.2019

Table of Contents

Participants:

Name	Affiliation
David Brock	Moderator, CHM, Director Center for Software History
Burton Grad	Moderator, SI SIG
Brad Allen	Inference
Avron Barr	Aldo
Denny Brown	Teknowledge
Ed Feigenbaum	Intelligenetics/Intellicorp; Teknowledge
Mark Fox	Carnegie Mellon
Peter Friedland	Intellicorp
Paul Harmon	Newsletter
Peter Hart	Syntelligence
Gary Hendrix	Symantec
Fritz Kunze	Franz, Inc
Doug Lenat	MCC/CyCorp
Brian McCune	Advanced Decision Systems
Peter Norvig	Google
Alain Rapaport	Neuron Data
Herb Schorr	IBM/ISI
Reid Smith	Schlumberger
Monte Zweben	Red Pepper/Blue Martini
Stephanie Dick	Historian, University of Pennsylvania (attended remotely using the Beam Robot)
Colin Garvey	Historian, Rensselaer Polytechnic Institute
David Grier:	Historian, George Washington University
Thomas Haigh:	Historian, University of Wisconsin at Milwaukee

David Hemmendinger	Historian, Associate Editor IEEE Annals
Hansen Hsu	Historian, CHM, Center for the History of Software
Dan'l Lewin	CHM, Executive Director
Paul McJones	CHM, Special projects
Len Shustek	CHM, Chairman
Ed LaHay	SI SIG, Meeting Project Manager

Burt Grad: What we're going to do is finish the first group of companies, and then we'll keep going and do some of the other companies. Again, try and stay at around 10 minutes or so. Some are a little bit wordier, like Peter, but you know, what the hell. We can live with that.

Advanced Decision Systems (ADS)

Grad: The next one on my list is Brian McCune with Advanced Decision Systems [ADS].

Brian McCune: Okay, I thought we were an outlier, but as I hear some of the discussions, I'm not sure we are. We were started late 1979. We informally spun out, meaning we walked out the door of a company called Systems Control Incorporated [SCI], which was in the Stanford Industrial Park. SCI in turn was an informal spin-out of SRI back in the late 1960s. So, it goes: Stanford to SRI, SRI to SCI, and SCI to AIDS, or ADS. There were two groups at Systems Control at that time. One group was doing a Hearsay-II system for signal understanding in order to track submarines.

Ed Feigenbaum: That was my project.

McCune: Ed and his wife, Penny Nii, started that project some years before that. That was the original impetus. Then, around that same time, my advisor, Cordell Green, left Stanford and talked to DARPA, whose funding was at the Stanford AI lab from 1972 through 1979. They agreed to just move the whole lab over to Systems Control Inc. So, we had two groups there doing program synthesis and doing signal understanding using knowledge-based systems, in particular the old Hearsay-II blackboard approach to that.

Four of us left SCI. The fifth founder was Edison Tse, who was from the engineering economic systems department at Stanford. We recognized that AI was hot, and that was 1979, but we also felt that it was really part of a much larger toolkit. Some of the techniques we've talked about here today, including control theory, database systems, operations research, estimation theory, decision analysis, and so on. These techniques, when put together in the right way, or oftentimes just one technique, allowed you to solve a very difficult decision-making problem.

Our whole goal was to have an R&D company, not to go public. As a matter of fact, not even to make money!

McCune: We wanted to create neat decision-making systems for real-world problems. We got a lot of funding from DARPA. We got a lot of funding from the rest of the Defense Department and the rest of the federal government. I think I've worked for every part of the federal government you've ever heard of. Then we also got a lot of funding from commercial along the way.

Grad: How much is a lot?

McCune: Now we were self-funded, by the way. Until we sold the company, we never took any outside investment in terms of equity.

When we started out, five people each put in $10,000, so $50,000. I think we actually sold too late in 1991 for about $10 million, $11 million. Probably two years before that we would've been worth $20 million, but the business model did change along the way

Grad: What kind of grants did you get? Were you talking $100,000? A million? How much did you get?

McCune: We had some that were in the range. We got one from American Airlines for $10 million. In our heyday, which was the late 1980s, we were getting contracts in the $5 to $10 million range for fairly significant systems.

Grad: Total revenue was how much? $20 million?

McCune: At the peak, it was a little over $20 million, not counting our spinoffs, which I'll come back to. The original model was an R&D engine that just looked like SRI basically, except we only did decision-making software. We didn't do hardware. We didn't do anything that was too simple—you know, the underlying tools, which are not simple. We just built, more or less, custom decision-making systems.

The first one, as I mentioned, was a system that Systems Control was doing, because I think it was so successful. You started at Stanford?

Feigenbaum: Stanford doesn't allow you to do classified research, so it was done at Systems.

McCune: It started out at Systems. It was classified, so it couldn't be done at Stanford. It looked very successful, so we thought, "Well, we can build lots of other things on this Hearsay-II model."

ADS' Initial Products

McCune: The first project we ever worked on, in fact, was something called the radar target classification system. The idea was you were supposed to look with an over-the-horizon radar at a ship like this, which is a US cruiser. What you were getting from the ISAR [Inverse Synthetic Aperture Radar] radar was this funny, fuzzy, smeared image like this. I don't know if you can see. We basically built a Hearsay-II system to tease out the different attributes from this thing. They were arrayed in a hierarchical ray, roughly, of knowledge levels from low-level signal

or image features to intermediate features like regions and so on, all the way up to the final hypothesis.

Then, at the top level, this system came up with a hypothesis. We had a library of all possible ships in the world, and you would do a match. At the top level, it was actually a model-based system. Because we had models of different ships and the different attributes, you should see if you were looking at that ship. We would just do this pattern match and rank those. Then we'd say either definitively "It's this cruiser," or "We're not sure if it's an oil tanker versus an aircraft carrier. You might not want to shoot at it. Because if it's an oil tanker, it could be a problem.

Grad: You were looking at it from over the horizon. You weren't looking at it down from satellites.

McCune: That's right. This was an over-the-horizon radar. Actually, in this case, it was on a point in San Diego Harbor, but the idea was they would mount it on a US naval vessel and look over the horizon to see who's out there before they shot their cruise missiles.

We reused this same basic idea probably two or three dozen times over a period of 12 years. We did not have generalized tools that we were using. It was really the architecture and a few tools, or pieces if you will, of what we called technologies of products.

Grad: What were they paying you to do?

McCune: These were applied research and prototyping contracts. It wasn't basic research, but it wasn't "roll out a product." There's a phase in government contracting that's all stuff in between, which is what places like DARPA do or some parts of the Office of Naval Research or AFOSR [Air Force Office of Scientific Research] that Ed ran and so on.

Hansen Hsu: Technically, this is 6.2 research?

McCune: It's 6.2 and 6.3.

Feigenbaum: Correction: I didn't run AFOSR. I was the Air Force Chief Scientist.

McCune: Oh, that's right. You weren't AFOSR.

Feigenbaum: I wouldn't run an AFOSR, but Peter spent a lot of time in the Air Force OSR.

ADS' RUBRIC

McCune: Okay. That was the basic idea starting in 1979. Then, at some place along the way, two things happened. One, the government came to us and said, "We're looking for tools to find relevant documents in large corpuses of text." My background in that was two-fold. One, at the Stanford AI Lab we had this system that Marty Frost had developed called the News Service, or NS System" It was strictly keyword-based. You typed in a few words, and it gave you some documents. We had the world's first real-time Associated Press feed coming into our PDP-10 complex.

That's the way I read my news starting in probably about 1973, so this idea that you don't need hard copy is sort of a very old concept to some of us. I knew about using simple Boolean keyword search to do what I would call standing or repetitive queries. These are queries that a knowledge worker wants to do, essentially, once every 24 hours or constantly be alerted from.

The second thing that happened is Ted Shortliffe, who did MYCIN and is a friend of mine and was my housemate at Stanford, did his thesis on E-MYCIN. So, when the government came to me and said, "Can you find documents?" I said, "The best thing to do is take E-MYCIN and build a News Service System around E-MYCIN."

That's what we did, and we called that RUBRIC. That was in 1981, and we continued to get more and more contracts in this area.

Grad: What were you selling again now in RUBRIC?

McCune: We were selling 6.2, but then 6.3 prototypes.

Grad: So, again prototypes.

McCune: That is prototypes where we would actually modify and then install the search engine into an existing government environment. There, real-time data feeds to it. Then we would do some level of an experiment. They could actually do better work, faster, with fewer people, than they could previously. It always worked, possibly because the alternative was so grim in those days.

We kept doing that, and then in 1986, we started a little skunk works. Their goal was to see, out of these 25 prototypes that we had been building for six years at that point, is there anything that's commercializable in a shrink-wrapped sense? RUBRIC was not the number one candidate. It was number two on the list. Actually, we brought in Bill Perry's company. Some of you may know Bill Perry Stanford; he was Secretary of Defense as well. They helped us prioritize this list. RUBRIC was number two.

Verity Spinoff

McCune: Our skunk works worked for a while on number one. I can't remember what the topic area was there, but it failed miserably. Technically, they just couldn't get it to work fast enough. They said, "Let's go on to number two." Bingo. They created the CIVERN].

RUBRIC was all on LISP, and of course, no client wants to see LISP product deliveries. It's like, "How do I maintain this thing?" So, we recoded it in C; made it much, much faster.

Then in 1988, we created a spinoff company and raised venture capital. That company was called Verity.

Grad: How much money did you get to start Verity?

McCune: I don't know.

Grad: Some of you are interested in numbers, but most of you aren't! It's interesting.

McCune: Well, I was involved in the creation of the company, but not in the financial details.

Grad: Were you a stockholder?

McCune: Oh, yes. Well, I wasn't. ADS was the stockholder. It was an odd model, because we owned a third of the company, and the venture capitalists were very upset that there was this extra thing here that was not a founding bellybutton employee. It was only a third of the company, and you know, once they got our technology it was like, "How do we get rid of these guys?" That company was quite successful.

Grad: That was a real product sale then?

McCune: Well, that's what we thought, but initially, it was not. It sold the product and services, and you really couldn't use the thing for a number of years without services. Then they created a shrink-wrapped version that ran on the PC. That was a 1993.

Grad: This is Verity now?

McCune: This is Verity. With that, they rode the Internet rise, and they went public in 1995 as an Internet company, even though they were barely a shrink-wrapped PC software product company. But that's my jaundiced view.

Grad: Were you directly involved in Verity or not?

McCune: Directly involved, I would say no. I was an OEM [original equipment manufacturer]. I used their product. I continued to license the product back to ADS, so I could incorporate it into all the systems.

Grad: So, you continued to do new services.

McCune: I was a user, a buyer of the product as opposed to a developer.

Peter Friedland: But didn't Verity continue to morph itself into the 2000s as a major internal search engine for lots of products?

McCune: Oh, yes, of course.

Friedland: Yes, they were a big company. I guess they got bought by somebody else.

McCune: Yes, they got bought out. It's a long story. They got bought out by HP [Hewlett-Packard], so now it's part of HP. Supposedly, it's the largest installed base search engine in the world.

Grad: Really?

McCune: Which is not saying much.

Grad: My last company bought Verity software as the internal search engine.

McCune: Oh, yes. That was our business really. That was the business. That's not what they went public on though.

Selling ADS to Booz Allen and Hamilton

Grad: So, what did ADS continue doing?

McCune: We kept selling systems that did multilevel fusion, multisignal fusion. For example, if you have two aircraft looking at another aircraft using different types of sensors, if they can share that data, you can figure out what that aircraft is a lot better.

Grad: What's the end of ADS? How does it end?

McCune: How does it end? Okay, so, as I mentioned, we had the skunk works starting in 1986. By 1988 we spun out Verity, and suddenly every person we hired wanted to do the next Verity. Why not? We had a core R&D group, but the core of it kept getting smaller and smaller, because nobody wanted to just do that. They wanted to go create Verity II. So, there was a basic tension that was never really resolved. We spun out three companies. Verity was the only one that was highly successful. We sold ADS in 1991 to Booz Allen and Hamilton, the international consulting firm.

Grad: What did they pay you?

McCune: It was around $10 million.

Grad: Not bad.

McCune: That was 1991. I think if we had sold in 1989, it would have been $20 million.

Grad: The difference between having a Republican and a Democratic president, is that what you're saying?

McCune: No, I think it had to do with the boom-bust cycle. AI was not as hot in 1991.

Friedland: It's not location. It's timing.

McCune: Yes, it's all timing. Absolutely. And I don't know how to time the stock market.

Grad: Brian, thank you very much.

AI Corporation

Grad: Does anybody here know anything about the AI Corporation?

Gary Hendrix: Well, a little. I mean, it was Larry Harris.

Grad: Tell me about it…

Hendrix: Larry Harris, I think, went to Dartmouth or he was a professor there. He did natural-language processing, and he created a product called Intellect that ran on IBM mainframes. It was a natural-language system. This is similar to what we had at Symantec, but his ran on mainframe computers, and I'm sure he had to go in and customize it for each new database that he used.

It was question-answering retrieval from the database. It was similar to what we had done at SRI with the ladder system, but it was just for IBM. He went with the big machines, and I went with the little ones.

Grad: Were they successful? They eventually merged with Aion to help form Trinzic. That is what sticks in my mind. Anybody know this or not? Ed has a complete blank in this area every time I mention that.

Hendrix: I don't know. He was on my radar, but he wasn't a competitor, because we were in different markets, so I didn't pay that much attention to him.

Grad: Okay. So, anybody else know anything about AI Corp? Did you compete with him, any of you? Did you use their products? Did you know their people? Total blank. Okay, they're gone. Take them off your list, Paul. They don't exist.

Paul Harmon: Well, AI Corp was bought by Aion, right?

Grad: Yes. I have no one here; that's one of the disappointments. There was a number of people who were really strong involved with Aion and in the merger with Trinzic. Harry Reinstein was one of the few people I knew in this field, because he had been involved in the trade association. He died, unfortunately, a few years ago. We went to four other people, and I got some notes about the company, but I'll bring them up I think tomorrow. Well, might as well finish it now, because it's not going to take long. I tried to get Scott Grinis. Anyway, according to this, Aion was founded in 1986. The principals were Harry Reinstein, Larry Cohen, and Scott Grinis; they were the three who started it. Do any of you know any of these people?

Friedland: Well, I knew Harry quite well. He was at IBM for a long time, wasn't he?

Grad: Oh, yes.

Friedland: I knew Harry from IBM days. More than Aion days.

Grad: I also knew him when he started the company, because he was part of the trade association [ADAPSO].

Herb Schorr: We had some interaction with him too when we were looking for products to license.

Grad: He got venture financing from Exel, Brentwood, and Asset Management, so he had some high-level financing.

Avron Barr: Did you talk to Jan Akins?

Friedland: Yes, I was going to suggest her also.

Grad: We tried to reach her. She's in New Mexico or somewhere. Wished us well. Everyone wished us well, but they wouldn't come up here.

Feigenbaum: I got an email the other day. She's in Tucson, Arizona. Loving it. Deer out the window. She loves seeing the deer.

Friedland: She finished up her career at NASA Ames actually.

Grad: Yes, that's what she said. It's unfortunate she couldn't be here..

AI Corporation Products and Services

Grad: Let's keep going. Principle products and services: They developed expert systems, and an advanced inference engine and object technology which were used by financial services and insurance firms to develop risk-scoring and underwriting applications. Were any of you competing with them in this area?

Peter Hart: My company sounded like it did, but we were actually in different markets.

Grad: Who were you selling to?

Hart: The largest banks and insurance companies, but we were doing risk analysis at what's called commercial level, so very large loans and very large commercial insurance policies.

Grad: That was why you talked to underwriters and people like that. Got you. Their major market was insurance and the revenues; they were in the three to five million range in the first couple of years.

When did they merge with AI? AI Corp and Aion merged in 1992 according to them. Who were the principals? This was Jim Gagnard and most of the Aion executive team went on with Trinzic. That was a name that Ed didn't even know, Trinzic; I was surprised. They did last for a while, I guess. The AI Corp CEO became chairman of Trinzic.

The business model: They sold B2B software using a direct salesforce in the US, and they sold to international markets. It was sold to Platinum for $240 million and then was acquired by Computer Associates International [CAI]. This was Charlie Wang's company.

This is an aside: Charlie Wang was considered the best garbage-collection software product company. Anyway, Charlie Wang considered himself the garbage collector, and when he bought a company, he got rid of 75 percent of the people and he never put another dollar into development. I have some biases in this area.

He made a fortune out of raising the maintenance fees when they were 12 to 14 percent to 20 percent. He'd raise them as high as they would go until people would stop buying. He built one helluva big company. His partner, his president, went to jail. Charlie did not, which is fascinating. Nor did Charlie's brother, who was the financial manager for the company.

Friedland: No one around here went to jail. That's another thing to point out.

Aion and the Mainframe Market

Grad: CAI still has the Aion product, still sells it as a CAI Aion thing. It's got to be one of the longest lasting products they're still peddling. But with CAI, nothing ever died there. Nothing ever died. Every product they bought lasted forever, and just as an aside, there were tons of utilities. Some of you may remember in mainframes. There were about 20-some companies selling utilities. They bought up 18 of those 20 companies, so they sort of dominated the utilities market. There's no price point problem, right?

Anyway, Aion at $240 million I think may be the high price for any of you on a sale.

Friedland: Well, wait. It wasn't Aion that was bought for that. I thought it was Trinzic that was bought.

Grad: Yes, but I'm saying that's the company; Aion and AI Corp became Trinzic. I don't know what else they bought. They did buy some other companies, but they weren't a major purchaser. Just as an aside, during that period of time, I was doing valuations for companies buying other companies. This was my job and Trinzic was not on the radar as being one of the big companies, but they did sell. Platinum was a big buyer and then CAI. I was consulting with Sterling Software, which was a major buyer. They bought 80 companies during the 18 years I worked with them.

Harmon: We really haven't discussed platforms very much, and it was very important in the market. Initially, some of the companies were run on DEC machines. Small software was run on PCs, but they were 286 machines at the time. Aion and AI Corp were the first companies

that targeted the mainframe market. They had it completely to themselves, because none of the other vendors were targeting the idea of running on a mainframe. It was a very significant market given that people were interested in AI, would try it on anything.

Grad: Exactly. Very interesting point.

Harmon: It paid off very well for Aion.

Grad: Anybody else want to make comments on either AI Corp or Aion or Trinzic?

Schorr: Harry Reinstein was an IBMer. That's his background. Harry came from the IBM world. From my perspective at IBM trying to buy product, they all offered me stuff on LISP machines, and I thought, "What the hell am I going to do with that?"

LISP Development Environment

Grad: That's something we haven't explored, Herb. Thank you, Paul, for bringing up this platform issue. Because most of you apparently were using LISP machines to start with. Is that a fair statement?

Friedland: We started on a DEC timeshare machine.

Grad: Then you went to the LISP. Why did you make that switchover?

Friedland: Because we wanted to lose money. $100,000 apiece.

Monte Zweben: I would argue though that the LISP machines were the best development environment ever invented. Even to this day, there aren't any IDEs [integrated development environments] that parallel that. The problem was it was a beautiful architecture that allowed you to debug and build wonderful programs, but it was specialized hardware using a tagged chip. To this day, I think if they would've focused on building the same development environment on a, say, Unix workstation that probably would still be a programming language that's popular today.

Doug Lenat: Yes, I think, if anything, you're understating things. We continued to develop on LISP machines until PCs were approximately 50,000 times faster and about 100,000 times cheaper. Anyway, when they finally got the 50,000x difference, then we regretfully switched to develop on "normal" machines, but they were astoundingly better for development.

Friedland: Arguably, Interlisp was the best development language ever made. I mean, I see people developing software today; the things we could do in terms of tracking programs and runtime and execute time, all the things you could do in Interlisp, they just don't do today.

Grad: Were you compiling the programs?

Lenat: They were compiling.

Grad: It wasn't an interpreter, because with an interpreter we could do those kinds of things, but they were so damn slow.

Lenat: When things sort of worked, you would compile them for speed.

Zweben: Right.

Grad: The development though was an interpretive process; that's what I'm asking you.

Lenat: Pretty much.

LISP versus Prolog

Harmon: The Japanese launched their fifth-generation program, and at the time, they announced they were going to be developing in Prolog. We haven't got into Europe at all, but the European systems, high-end expert systems, were all Prolog-oriented, because they were going to support the Japanese. The United States was LISP-oriented.

There was a movement in the United States to develop LISP chips, chips that would be LISP machines, as opposed to running LISP on a PDP or something like that. On machines that were specific for that purpose. I mean, it just ran its course.

Grad: Why was LISP so slow?

Harmon: It's interpretive.

Lenat: Well, let me answer your previous question. It's not that this was a particularly better language than Prolog, although that's another whole discussion since I think it was. The differentiation was the astounding amount of brilliant time, effort, successful development that went into the Interlisp and Symbolics LISP development environments. Developing programs in those languages, you were surrounded by an environment that made things go so much faster. When you had to debug, things were so much faster.

Zweben: Totally.

Lenat: That's what made it better.

Brad Allen: It was all the same thing from the screen down to the silicon. It was all LISP. It was pure and elegant.

Zweben: Yes.

Allen: It astounds me that to this day we still don't have a system that works as well and as beautifully as that thing.

LISP Runtime Environments

Friedland: Burt, let me answer one of your questions about delivery environments. In 1985, realizing that not everyone was going to have a Symbolics machine or a Xerox D machine or a Texas Instrument's Explorer, which were the three main LISP machine development platforms, IntelliCorp actually went to sort of a hybrid approach. If you were doing development, if you were going to buy Key and develop systems, you had to have one of those machines, but it had a runtime executable that would run on things like VAX simulators. Remember, in that period, you were transitioning from timesharing machines to mini-computers, VAX simulators, and machines like that. That was their approach to selling into the runtime environment of corporations.

Grad: I'll go back to my question: I can understand the interpretive, but once I compiled and said I'm in execution mode, why couldn't it run fast?

Friedland: It did. It could.

Zweben: What came out was a RISC architecture. What happened was that it was built on a tagged chip that was specific to LISP, as Brad said, and it was very good for a long time, but Reduced-Instruction-Set Construction computing came along that started to supersede the performance of the tagged architectures. No one built the development environment that the LISP machine was built on for a RISC architecture.

Grad: So, it wasn't a big enough market.

Zweben: I think that was religion. That wasn't market.

Allen: Yes, it wasn't. Dick Gabriel started a company to sell the LISP specifically to the workstations that were emerging. I forget the name of that company.

Friedland: Well, there were LISPs for the workstations, but no one developed the wonderful thing that we're all drooling over: the Interlisp-type development environment. As Doug said, LISP itself was a trivial language: a left parenthesis, a function, arguments, and right parenthesis. That's it.

I took McCarthy's course on LISP. That's what he told me, "This is the course." I still keep that next to my desk because it's such a wonderful thing. It's that thick. That's because people like Larry Masinter at Xerox PARC eventually developed these just beautiful environments.

Lenat: And Warren Teitelman.

Friedland: And Warren Teitelman, of course. Yes.

Unidentified: Is that the LISP D environment?

Lenat: Yes.

Friedland: Eventually, yes.

Allen: That brings up an interesting thing, which we'll probably get back to at some point.

Grad: From a marketing point of view, most of the vendors who were here originally came out of academic environments and supported LISP in the United States. Other vendors who wanted to succeed in the market were supporting other languages, because the business people simply didn't understand LISP and they didn't want to learn it. They certainly didn't want to learn it to run it on mainframes, for example.

LISP Programming

Schorr: At one point, TI came to me in my IBM role and said they wanted to do joint marketing and would sell the TI Explorer, to IBM. I ran into this $100,000 a pop. We already had Sun workstations and RISC and so on. I said, "This is non-competitive, and I'll tell you the second problem." I said, "Give me some performance metrics as to why this outperforms the Sun workstation." They sent me to MIT. What's his name? Richard Greenblatt?

Friedland: Gerry Sussman?

Grad: Go ahead with your story.

Schorr: There were no metrics. They couldn't show me anything nor did they know how to make a case about the performance. Maybe it was a better programming environment, but I had another problem: I had Cobol programmers. I didn't have people who could program in LISP.

Lenat: There's a cure for that today.

Schorr: Not much. Cobol is still very strong. You may not believe it, but it is!

Friedland: We got Explorers for 40K each.

Schorr: They're very good.

Allen: No, listening to Peter, maybe something we might want to touch on at some point is the differentiation that was made at the time between East Cost AI and West Coast AI.

Friedland: Sure.

Allen: That was an interesting thing, which kind of came out of this. I mean, Interlisp was a big thing out here. The Xerox environments back east were MIT and Symbolics and Maclisp and so forth. There were these kinds of ideological battles that people would have, which were really basically over these minutiae of which development environment was best.

Grad: We've gone off topic, but we'll do one more thing, and then we're going ahead.

Fritz Kunze: I understand from Burt that he's going to let me talk tomorrow after I've had some sleep. So, I'll give you a more interesting version. In 2008, I gave a talk on the history of the LISP wars at OOPSLA [Object-Oriented Programming, Systems, Languages & Applications] and I was videotaped, but fortunately my slide show had a bunch of images in it, which I could not get permission from the book publisher to include. So, I didn't sign over the rights to make the video.

Most of what these guys are saying is right. There was a big bias between the East Coast and West Coast. People from MIT thought the Berkeley guys were just not very good. And we were sort of pissed off at them. The problem was, at the end of the day, the people that built the Symbolics LISP machine environment were really smart, because the old guys were much smarter than the new guys. I have a theory about that, too.

I have a whole slideshow. There's a really great theory about this, but the bottom line is we couldn't replicate it, and his point is right. If you had to teach Cobol programmers, it's very different to do it, because there's beauty in the environment that once you got the metaphor for it, it made perfect sense, but you couldn't get the metaphor easily. You know, you had to live in it. You had to be steeped in it, basically.

Grad: Is this a subject we should spend a little bit of time on tomorrow?

Friedland: Or it could be the 16th Pioneer Workshop.

LISP Environment versus Machines

Reid Smith: Let me add a little bit of perspective, at least from the viewpoint of one company that was pretty big in this area. We developed eventually the Dipmeter Advisor system, which was the first of the commercial systems that Schlumberger offered in oil and gas and Interlisp-D. There's the elegant LISP-D part of the environment, but BITBLT was the thing. You remember? It was the graphics systems that made a huge difference, and it was integrated totally with everything else. That's great.

Now you're going to deliver one of those to 150 countries around the world, none of whom, if you were lucky, had local networks, let alone anybody who might be able to support such a machine. It just wasn't going to happen at scale and never happened at scale until people rewrote all this stuff in C++ and could run it on VAXs or whatever it was in the day.

Brad brought this up, but I think this issue is the scale problem. Platforms I think are a big deal, and I think one of the things you can ask is, what's different today? We're talking 40 years, 50 years. What happened? What's different today?

I think one of the things that's different is hardware became a nonissue, relatively speaking. It was an issue in the day. I was teasing Herb that in the day we were trying to do AI without the benefit of computers. I think I offended the group. I'm sorry. But relatively speaking, we didn't have much computational power or storage and no internet, no connectivity to speak of. It's a huge change over the years.

Harmon: For the historian, I think a lot of credit needs to be given to the expert systems people, who are actually educating American programmers. I taught lots of courses, and I don't have a strong background, but hundreds of people at IBM subscribed to my newsletter. I taught hundreds of people at TI. These guys, I think, started as programmers somewhere and learned programming in a very straightforward way, how to program in Cobol, and then they became programmers.

Then, all of a sudden, they were being asked to learn about expert systems or AI, and they really had no idea. I mean, they asked very fundamental questions about LISP and languages. In effect, expert systems, whatever else it did, actually got them to think about what computer science was all about in a way that many of them hadn't thought about before. It was a big step and they weren't going to leave Cobol and approach LISP. For most of them, it just wasn't going to happen. But it affected the whole expert system market because they were trying to sell into a population of guys that just weren't ready for it.

Haigh: In retrospect, do you think your companies would have done better if no one had ever had the idea of building a LISP machine and you'd have had to stick with DEC for a bit longer and then gone to Sun or Apollo and then gone to RISC, and not have taken that enticing detour into this system, which you clearly love, but it's also for an outsider hard to see that the idea ever made any sense for anybody?

Hart: The answer's no.

Grad: The question is, "Is it the language or is it the machines?"

Hart: He had better machines. Our company delivered these very high-end enterprise software solutions. Herb may be the only person in the room who remembers CICS on mainframe.

Schorr: I do remember!

Grad: I owned CICS. I didn't write a damn bit of code, but I owned it as a development director.

Hart: Anyway, our fate was unrelated to LISP machines.

Friedland: Just to answer your question, Tom, I would say that if somebody had developed a very usable programming environment around Fortran, it wouldn't have mattered to me that much in terms of development. LISP was a great language, but it was the environment for producing stuff in LISP that was important for me, at least in developing software.

Haigh: Yes. Was there any technical feature of the LISP chip that made it easy to have a good development environment, and could they have done the same IDE for any other platform?

Grad: Fritz?

Kunze: Yes, basically the LISP machines were tagged. What that means is that every piece of data in the machine is being processed; you could tell what was in that memory location. Is it a floating-point number or an integer? From that, you could easily debug programs. But the RISC architecture and the standard architecture were not tagged, so people building a LISP system had to build tags on top of that, carry them around, which slowed things down.

Grad: That's very interesting that the environment was really what was important, not the language per se. Is that correct?

Several: Yes.

Schorr: There was a fundamental problem. When you do a special-purpose machine, you spend a lot of money to develop that and there's not enough user base. It inevitably had to lose out to some workstations. These companies were too small. You're talking 400 to 1,000 people developing these chips so, it's impossible for a small batch to do it.

David Grier: I came to LISP machines because as a child I had been raised on the Burroughs B5000. It's the same issue. It's the environment. They tagged all the data; you could trace it, and it made perfect sense to me. Of course, Burroughs couldn't maintain that environment without shifting to a more conventional architecture, which lost the entire benefit of it and it collapsed.

To me, the killer of the LISP machine partly was the government contracts advantage. You could put that out for just a fraction of the cost, and you now had a whole bunch of open-source tools. It wasn't worth the difference.

Grad: I am stopping this conversation. We will pick up some of it tomorrow.

Carnegie Group

Grad: Mark, I'd like to have you talk about Carnegie Group.

Mark Fox: So, the genesis story: started with the DEC VC [venture capital] executive coming to Jaime and me to combine his natural-language understanding technology with my knowledge-based and relation technology. It was too early for me to do anything, given I hadn't even started writing my PhD at that time. But we got to talking and we brought Raj Reddy and John McDermott into the Sunday morning conversations. That started in 1982. At that time Raj just went off and incorporated the company because that's Raj. Raj Reddy, without us knowing, comes back *fait accompli*.

John McDermott had very strong ties with Digital Equipment Corporation, Dennis O'Connor, and I forget the VP's name. Anyway, given John's stature within Digital, because of his success with, not Sam Fulton. It was somebody else. But Sam was another person.

Anyway, Digital heard about it and came to us and gave us an offer that we couldn't refuse. They said they would give us a million or two of contract, plus a million in equipment as equity, to start the company. You have to understand at that time we were meeting, and we were toying with the idea of starting the company, but we didn't have any serious plans for anything. We had no business plan; we had nothing. We thought maybe we'd do some consulting and that would be it. Suddenly, we were handed a client on a silver platter. Then we had to reverse engineer a business plan because they wouldn't sign it until we had the business plan. So, that's how Carnegie Group started.

John McDermott, Jaime Carbonell, Raj Reddy, and I moved in. Now what's important here about us is that we're all very different in terms of the AI that we do. John is an expert systems person, I'm a constraint search planning-scheduling type person, and Jaime's a natural-language-type person. Raj Reddy is anything he wants to be, but his background was speech at that time. That turned out to be important in terms of the focus of the company going forward; and it's important at the end of the story.

In 1983, once we were incorporated, all of us around this table were being asked to give talks on AI at various companies. Inevitably when I went to Ford and Boeing and Texas Instruments and these other companies, we would have lunch and they would say, "Mark, what's going on? Anything new going on in your life?" I said, "Yes, we just started a new company." I told them Digital invested. The first thing they said after that was, "Can we invest?"

Between 1983 and 1985 approximately, Texas Instruments followed suit and made an equity investment and a guarantee of $2 to $3 million in projects. Ford did the same thing at a higher level of guaranteed projects and equity investment. Boeing Information Systems made the same deal. Compagnie General Electric from France did the same deal.

Carnegie Group Initial Products and Services

Fox: Suddenly, we had a backlog of millions and millions of dollars of projects, and we had sufficient equity to move forward. Now, they were interested in everything that we as faculty members did. They wanted to do logistics-related stuff, manufacturing-related stuff. They wanted to do electronics, aerospace—you name it, everything was what they wanted—and we had no products. We did not take any technology from CMU [Carnegie Mellon University] at that time. We just were the four of us, and everybody but Brad was there.

All the work we were doing at the outset, everything we were doing, was services. We were doing custom projects for our clients. And a lot of what the clients were interested in was not necessarily "We have this problem. It needs to be solved. You better deliver a product." It was, "We want to explore the application of AI within our own organization. So, we want to do some project to see how well it works to evaluate the technology." It was things of that nature.

Grad: This is commercial rather than scientific-technical-medical. Is that correct?

Fox: We were not a R&D organization for the DoD [Department of Defense] or for anybody else. They were evaluating what we produced from a business perspective. Now that doesn't mean that there weren't some projects that were actually delivering specific solutions. There was some of that in the early days also.

Now, let me go down the technology path and then come back to the company. From a technology point of view given my background in knowledge representation, simulation, etc., and given Jaime's background in natural-language understanding, we began to produce products. We didn't take the technology from CMU. We started from scratch.

We created a number of technology platforms. One was Knowledge Craft, which was a competitor for ART and Teleport's product, etc. It was a platform in a sense, and it provided a variety of tools and environments in which you can construct applications, in which there was knowledge representation, a frame-based language. There was rule-based reasoning, there was Prolog-type reasoning. There's a variety of things in there. We had something called Simulation Craft, which was a knowledge-based simulation environment. We had Language Craft, which was a natural-language interface development environment. —

Fox: Let me stop with those, and I'll get to follow-on products after that. We started all that, and we had a number of companies who bought our products, including IBM, but somewhere around 1985 we were running out of money. How can you run out of money when you've got a huge backload in the millions and millions of dollars? You have all this investment money. We were running out of money because we hired a CEO who liked to spend money. Didn't know how not to spend money.

Grad: Can you think of his name please?

Fox: Larry Geisel. Anyway, when we discovered he was running the company into the ground, I had to go in and fire him. I had to take over the role of CEO. So, I was the founding CEO and then I transitioned to VP of Engineering as we found the CEO. Then I had to come back and take over the reins of the company again.

Grad: You picked the guy yourself? It was your call/

Fox: If you want a little bit more detail about it, there is a Harvard case study on this thing. My memory jogged for what happened way back then.

Anyway, I came back in and I had to do a few things. One is I had to turn the company back into profitability. I had to find another investor. I had to raise morale of the company. I can't remember, but there's a fourth one that's listed in the Harvard case study. In a period of about nine months, I did that and I had to find a new CEO—that's the fourth.

We found a new CEO. We were profitable. We found an investor, U.S. West, and everything was good from that point on. We were a profitable company.

Carnegie Group Growth and Revenue

Grad: How big did it get during that next period there?

Fox: We grew to 300 people, and we grew to over $30 million in revenues. Let's see, what else? We went public somewhere around I think the 1996 timeframe, I think we went public, and we sold in 1998. That was 1985, I think, when I took over. In 1991 I moved back to Toronto, so my interaction with the company was strictly at board level as opposed to directly involved in projects.

But during that period of 1985 and onward, we developed some problem-specific products. One was S Bench, which was a diagnostic system for machinery, which was used by a variety of companies out there to diagnose imaging, like MRI [magnetic resonance imaging] machines, CAT [computerized axial tomography] scans, and stuff like that. We developed the Construe System, which detects categorization, which Reuters deployed to automatically categorize news stories. We also built KBLPS, the Knowledge-Based Logistics Planning System. Turned out that the Army became a large client of ours in the logistics area, so in-country logistics was being done by our KBLPS software and was taught throughout their military colleges and everything else like that.

Grad: Did you consider all of these expert systems?

Fox: I considered these knowledge-based systems. KBLPS was based upon earlier work I did in the scheduling area using constraints and what was called "constraints heuristic search." We also built a system that the DoD used for their medical transportation system, which looks at how you move injured people from one place to another where you have to marry up the person, the nurses, the doctors, the equipment with the aircraft to get them all from point A to point B. That was being done by Carnegie Group software.

Decision to Sell Carnegie Group

Fox: The company was growing. As I said, it grew to revenues north of $30 million, but we reached a point in 1997 where we lost confidence in the management team. The CEO was Dennis Yablonsky, and the reason we lost confidence was not because we were losing money or anything of that nature. It's just that we didn't see it growing as fast as we would like to see it, and we didn't see the leadership team as providing the appropriate leadership to take us to the next level.

Raj and Jaime and I sat down and said, "Okay, who's willing to come back and take over the reins?" They all looked at me, and I said, "I live in Toronto now. I'm not coming back." So, we decided to put the company up for sale, because we were unwilling to step in.

Grad: Were you still significant stockholders that you could control the destiny?

Fox: The outside stockholders had somewhere between 50 and 55 percent. We had enough to affect what was going on.

Grad: Interesting. It's an interesting story you're telling.

Fox: So, no venture capital. It was all companies who had a direct interest in the technology that we were producing and from a board point of view were interested in our success, but also interested in what we were producing.

Grad: There's a very different financing model than most of the companies used, and that's a very interesting strategy. Do you mind telling us what you sold it for?

Fox: I think it went for about $15 million.

Grad: Mm. So, it must not have been at the $30 million range anymore.

Fox: No, it was. This goes back to the point I was raising right at the beginning about the diverse interests of the founders. As a company, we did not have the focus, I think, that

would result in a higher valuation. That is to say we were in natural language, text categorization, diagnostics, manufacturing, planning, and scheduling, military logistics.

Grad: You were all over the place.

Fox: We were all over the place. We were not viewed as a product company per se, even though we had products.

Grad: You weren't a pure play.

Fox: We weren't a pure play. And that's reflected in the valuation.

Grad: Getting one-half times revenue for a software products company in the late 1990s before everything went to hell was a little low.

Fox: Yes.

Grier: Just for the back story, I'd like to know about the Harvard case study. Did you ask for it? How did you get chosen for it, if you didn't?

Fox: How did they do it? They approached us.

Grier: Okay, that's the question.

Friedland: It's basically a study of the first part, right?

Fox: Yes. It goes up to 1990. The Carnegie Group. Don't ask me how it started. I can't remember.

Teknowledge

Grad: I have two more in the early group. Then we'll have finished that and we'll get done exactly at five o'clock. Denny, you go first.

Denny Brown: Teknowledge was formed in 1981 by 20 founders, five or six of them in the room. Avron, Ed, Doug, Peter—I think that's it in the room. In the early piece, it was focused on education and training. That was the visible part.

I went over in January of 1982 after leaving Stanford. Paul Armer and I changed jobs. Paul had been working for Teknowledge for a little bit. He took the job that I'd been doing at Stanford, and

I took the job at Tek. As I said in the intro, we were doing a one-day executive briefing, a one-week overview of expert systems, a five-week intensive that was parallel to the MSAI [master of science in artificial intelligence] program, core AI part, and then a six-month training program. At high water, at one point we had 24 people in the six-month program from all over the world. So, that was the initial work.

Early in 1982, the founders, the board, however, hired Lee Hecht to run the company, and he started with the same kind of approach that Mark was talking about with getting funding from specific companies—General Motors and others—and contracts, but for a while some of the revenue for the company was coming out of the education and training aspect.

Brown: You're going to ask me the question about what did we sell.

Grad: You've been listening to me!

Brown: We started out selling the education and training programs. Relatively quickly, the management wanted to do an expert systems shell. We were in the MYCIN/E-MYCIN historical. E-MYCIN became what we called KS300, which was a redo and extension of E-MYCIN, which then was marketed as S.1. S.1 was the name of the tool. It combined a little bit of some of the object-oriented stuff, but it was mainly a rule-based system, like E-MYCIN. From a genealogy perspective it was E-MYCIN with a bunch of stuff around it. Meanwhile, while we were bringing S.1 out, the interesting thing with Tek was early on bringing in people that didn't really have the expert systems background to run the company.

Lee Hecht hired Barry Plotkin. Those two were running a good deal of communication between the business side of the equation and the technical side. We basically started to do some experiments on how were we going to make money. We had the education and training. We had the product business. We had some government R&D work that was going on. Then the building of custom applications became the biggest money-maker. In the 1984, 1985, 1986 timeframe, we were doing $25 million a year, something like that…

During one of the reorganizations, we reorganized into three main business units. That really was an experiment of, what's going to win? We had a products group that was products and training. We had the custom applications group, which was build custom applications for the big companies, and we had an R&D group that was doing government contracting.

Grad: One question: In building the applications for people, were you using your own tools?

Brown: Good question! That's an excellent question. We were and we weren't. Some of the applications were being built on S.1, and some of the applications were being built on

technology that we were developing in-house with more of the object-oriented flavor to them, including still the rule-based technology, but with an object-oriented and different development environment than was happening with S.1.

Grad: The next question is financing?

Brown: Yes, the financing. It was mainly investments from the big companies.

Grad: Was it investment, or did they buy services from you?

Brown: Both. Some of it was contracts, where we would come in with these projects, and some of it was equity.

Grad: Ballpark numbers?

Brown: Don't know. I was not at that level, and that wasn't important to me at the time.

Friedland: I have the numbers here for you if you want.

Brown: Oh, yes. Peter brought this.

One of the things to note with my piece of this story is toward the end of the run,1987, toward the end of 1987, they fired my boss, who was running the custom applications group at the time, and he formed another company. I gave my notice at Tek and they found out that I was leaving to go there, so I was walked out. So, I have no pieces of paper.

Friedland: Really! Oh my God.

Grad: What was the other company?

Brown: Coherent Thought. Coherent Thought's not on your secondary list, but it could be.

Grad: And who was the person?

Brown: Barry Plotkin was the person that they fired from Tek and who started Coherent Thought. It ran from 1987 to 1990, building custom application development tools actually for the mainframe. We were headed to CICS when the net went under.

Grad: CICS?

Brown: CICS is the transaction processing system for mainframes. We were headed there, basically taking the architecture and the development environments from the LISP world, building them in C, and running in an environment that would go into the CICS environment. That was the direction. It was venture-funded, and shortly after we formed the company, Teknowledge sued us. That lowered valuation on the first round. We went off our second round at Coherent Thought in 1990 when Saddam invaded Kuwait. Money dried up, and the VCs pulled the plug.

Selling Teknowledge to American Cimflex

Grad: What happened with Teknowledge after they walked you out?

Brown: Teknowledge went public in 1986. Peter has the pieces of paper.

Friedland: I'm the packrat of the group.

Brown: So, public in 1986. In 1987 they fired Barry. That really crippled the custom applications group in ways that they didn't recover from very quickly. By 1989, they sold themselves to American Cimflex. American Cimflex was a company out of Pittsburgh?

Friedland: Yes.

Brown: Pittsburgh, I think. This was after I was done there. American Cimflex wanted to go public, and at that point, the IPO market was depressed. Teknowledge had money but didn't know what they wanted to do with it. So, it was a merger where American Cimflex took over majority ownership.

Friedland: It was one of those strategies where a private company became public by acquiring a lesser-valued public company. You were probably aware of that.

Brown: Anyway, that's what happened.

Teknowledge Corp Spinoff and Sale to Intuit

Brown: To continue the story, which I only know by reading the books, in about 1992 or 1993, somewhere in that neighborhood, they spun Teknowledge Corp back out and Neil Jacobstein and Rick Hayes-Roth ran the Teknowledge Corp. They got themselves into two things really: financial services and government contracting.

Grad: Applications, though?

Brown: Yes.

Friedland: They became a DARPA contractor. It was their main business.

Brown: Main business was DARPA. Yes.

Friedland: They became part of DARPA teams and were very successful at that for a while.

Grad: Could make some money.

Brown: Yes. Then they did some financial services, custom work. Intuit bought them, right?

Friedland: That's past my knowledge.

Feigenbaum: When did they run out of gas?

Friedland: When did Intuit buy the financials of Teknowledge?

Brown: It was probably late 1990s. mid- to late 1990s.

Friedland: Intuit bought part of it, and then the rest lingered on as a DARPA contractor. When did it finally go out, Ed? Do you remember?

Brown: Well, the bankruptcy was 2013. I found that out last night.

Friedland: Seems like last year.

Brown: December 2013.

Grad: I think you brought it to a close, Denny.

MCC and the NCRA

Grad: One more. Doug, you're going to be the one to talk about MCC.

Lenat: Burt has wisely called on me last because he knows I have a 5:30 pm meeting that I have to attend.

Grad: He has to leave at five o'clock. I didn't plan that!

Lenat: That's okay. MCC is an outlier in a lot of dimensions, but I think it is still worth this group's time and this camera's time.

Grad: They weren't a real company in the sense we're talking about, were they?

Lenat: They were, sort of. I guess originally there was a problem and opportunity. Let me talk about the opportunity first. The opportunity was that America was scared to death of the Japanese Fifth-Generation Computing effort, which threatened to do in computing hardware and software and AI what they had just finished doing in consumer electronics and the automotive industry. As a result, Congress did something, which is how you can tell it was 30 or 40 years ago.

Mainly, they passed something called NCRA, the National Cooperative Research Act, in 1984. It was sort of drafted about a year before that. What that said was, "Hey, all you large American companies, normally you would be violating antitrust laws if you colluded on R&D, but we promise that for the next 10 years, none of you will get sued for doing so as long as IBM isn't one of your groups."

Grad: Really?

Lenat: About 20 large American companies, not including IBM.

Friedland: They named Herb Schorr in the legislation, right?

Lenat: Specifically. They came together and essentially pledged to each contribute a couple million dollars a year. That added up to a decent amount of money for 10 years to fund high-risk/high-payoff research in a few areas that were supposed to fit together, one of them being artificial intelligence. The idea was do research which may take five, 10 years to pay of—maybe even a little bit longer—but which if, it pays off, will help America's competitiveness on the global stage. It's high-risk, high-payoff, long-term research projects.

Bob Inman came to see me at Stanford and essentially said, "We're looking for principal scientists to sort of coordinate these things." This gets to the problem: just at that point I'd been talking loudly, along with Alan Kay and several others, about what we saw as the impeding failure of expert systems to scale up for various reasons involving things like the difficulty of maintaining a consistent knowledge base. As the number of rules goes up, you don't want the time to enter "rule *n*+1" to keep going up with *n* and things like that.

As a result, I went and did that for 10 years, and it was a very exciting period of time. We got a lot accomplished. We actually did solve the three or four big problems that I saw that expert systems were having. We uncovered approximately 136 other big problems that I hadn't thought of that expert systems had, and we did actually make a lot of progress on those. I can talk more about that maybe tomorrow when I talk about Cycorp.

MCC Investment and Revenues

Lenat: The aim for MCC was to make money for the investor companies by developing technologies. Some of the revenues would flow back to them by having spinout companies like mine, some of whose revenues would flow back to them, and so on. It was, in a sense, a company that was trying to make money off of AI and expert systems along with a few other things as well.

Grad: Doug, it took 20 companies about $2 million each year?

Lenat: Yeah, $2 to $3 million.

Grad: So, you had $40 million.

Lenat: $40 to $50 million each year. Yes.

Grad: Okay, thank you.

Lenat: A lot of the reasons I wanted to mention it is that there's a common misconception that MCC was funded by government money, which is not the case. I think partly to blame for that is the fact that the CEO was Admiral Bob Inman, who'd been at NSA [National Security Agency] and CIA [Central Intelligence Agency] and, of course, was an admiral in the US Navy and so on. He came from that world, but MCC did not take government money.

About 10 years later, starting in the mid-1990s, American companies became less afraid of the Fifth-Generation effort because it was sort of crumbling and failing. They became less American companies and more multinational companies. They became more afraid of mutual fund managers than of the Japanese. So, for various reasons, the source for funding was drying up.

MCC then began to take some government contracts in its last few years of life. But the most successful projects, including the Cyc project, spun out at that time. We spun out at the end of 1994.

Grad: What was your role, Doug?

Lenat: So, I was the principal scientist at MCC.

Grad: Did you actually run projects there?

Lenat: Yes, I started and ran the Cyc project, which most of you know about and I can mention a little bit more tomorrow. Essentially, this goes back to "What was the problem?" One of the problems with expert systems was that the incremental cost to build each expert system was pretty high, partly because no one was taking the trouble to codify the millions, maybe tens of millions, of rules that all of us have and that in some sense all expert systems should start with, instead of starting with nothing. Once that mattress gets pulled off of the road, traffic should be able to advance, and the incremental cost to build an expert system might drop by one or even two orders of magnitude.

Value Propositions of Expert Systems Companies

Grad: It's an interesting point that he raises. Many of you were building applications. Did you find the cost of building them basically stayed the same over the time? There was no improvement in your productivity?

Friedland: Yes, as the environments improved and as the speed of hardware improved.

Grad: What was the execution time? I'm talking about the building.

Friedland: No, no, no. As the true building environments improved, the time to build systems became less.

Grad: But he's saying that wasn't his experience.

Lenat: A lot of strategy was reuse. People would build something, and then as you said, reuse it, reapply it, re-customize it. Effectively, it was almost the same expert system and almost the same application, five or 10 or 20 times.

Grad: So, there was an improvement. Denny?

Brown: But Doug's exactly right that these were expensive. The custom applications that we built at Tek were expensive. There was a lot of system integration issues that were being addressed. It was very human-intensive.

Grad: Let me stop you for a minute. What puzzles me is the payoffs. When I think about Schlumberger, some of these payoffs were incredible. They mentioned somewhere—I guess

maybe it's in your book—a United Airlines thing where there was $100 million worth of payoff. You had payoff, Herb. Many of the projects that were done in IBM that were very significant, big money was being made. You would think the money was enough; they would pay you anything to do it.

Brown: Right, but with big investment up front on a technology that wasn't "proven" yet, big companies would build these things and some of them would pay off.

Grad: Let me tell you what I'm wondering. Was it so easy to use the technology that they could do it themselves, they didn't need you anymore?

Brown: Uh-uh.

Lenat: No.

Friedland: Wait a minute. I don't disagree as much as some of you guys do on that question. I think the reality is sort of what people show you their heads on. To go back to Bioinformatics, one of the reason why Bioinformatic Chronics as a company tended to dry out a bit was the perception that as they became important to the companies themselves, the companies decided rather than having some small company do it for them, they would bring it in-house. That happened a lot in the expert systems industry.

Grad: But that's what I was wondering.

Friedland: You saw companies like IntelliCorp lose market in essence to the people who had come to them originally for help in building systems.

Lenat: Like Broling.

Friedland: Yes, exactly like Broling, who built a very big expert systems, knowledge systems group themselves, because once it was important to them why would they rely on some dinky little company to do it for them, as opposed to doing it themselves?

Grad: The answer always is that the technology, the software, that that company has, whether it's Peoplesoft, SAP, or somebody else, is worth getting from them, because they have something that you don't have. Anyway, let me let Doug finish his story.

Grier: I've had multiple questions. First, on that one, to what extent was it exposing your knowledge to an outside firm? That you bring it in-house, you can keep trade secrets.

Harmon: I'll answer one of your earlier questions: was it software they were selling or was it consulting they were selling? Clearly, they weren't making enough money on software. Once companies decided they didn't want to have their consulting…

Grad: It was lousy pricing, right?

Harmon: Too small of a market for software.

Grad: Doug, please finish your story about MCC.

Lenat: Okay, I'm going to mention one other factor in case there's anyone in the room that I haven't antagonized yet. I think another serious factor, and I mean this very seriously, is that in our desire to share our excitement and share in the fruits of applying expert system technology standards got lowered and the people who we were training, for example, at Teknowledge were not of the same caliber as the people who had, say, built MYCIN and so on.

My archetypal example of this is someone who is otherwise a very smart human being who was mentioned earlier today: Lenny Forgi. When I got to see him, you showed me an expert system he built, which essentially was something that had rules like, "If program counter equals 804, then *X* becomes *X*+1 and..." Basically, he was programming in Assembly language, but using an "expert system" shell in order to do it. It's like, "Oh, my God, you've missed the point of expert systems!"

Friedland: Go back to Fortran.

Lenat: This applies to almost all of us in this room, including me. I don't want to say "greedy," but let's say we were over-optimistic about things. In effect, we ruined the space by diluting it with people who then went back to their companies or were at their companies all along, built systems which didn't really perform very well. That basically created a bad taste for expert systems in the mouths of a lot of the companies. So, I think that was one of the problems.

Grad: Finish your MCC story.

Lenat: For reasons I mentioned, MCC began to spiral out of existence. It finally went out of business in the late 1990s. I think it officially declared bankruptcy in the year 2000. One of our employees who was smarter than I was went to MCC just at the right moment and bought out their share of CyCorp. He's now like the second largest shareholder in CyCorp, but I still have most of the stock and we'll talk about CyCorp tomorrow a little bit. But MCC, essentially, closed its doors in the late 1990s.

MCC was based in Austin, Texas. Almost every large city tried to vie for it to be located there. It was located in Austin for a variety of reasons, one of which was the University of Texas provided a brand new large rent-free building for 10 years for MCC. The governor gave up his private jet to fly MCC recruiters and senior people around the country and banks were somehow in on this. From the time that we found the house that we wanted until the time we moved in was about 48 hours. In general, Texas had a tremendous package to offer, in addition to there not being any state income tax in Texas and real estate prices were drastically lower even in those days than they were on the East Coast and West Coast. Texas was actually a very nice place to settle.

When we spun out, we could have actually relocated anywhere, but all of us had fallen in love with Austin, Texas, by the end. We all decided to stay there.

Grad: Of the companies that were there in MCC, were major ones in Texas?

Lenat: TI was a member, but most of the companies were not particularly Texas-based. For example, by the time we left, Apple and Microsoft had joined and were supporting our project.

Grad: It wasn't the fact there were three or four big companies in Texas.

Lenat: No, no, no. Not at all. These were large companies with a bright future, like Kodak.

Grier: Doug, do you see a connection between the MCC story and the enabling legislation for it and Symantec?

Lenat: Yes, Symantec formed, I think, a year or two later. I don't remember the exact timing, but Sematech was much more focused on the hardware and chip end of things, but again, it was a large consortium. There were a few other consortiums that formed thanks to the NCRA passing. I don't know the Symantec story though.

MCC and the DoD

Grier: The connection in MCC you were reporting or your government manager, if you will, was part of the Department of Defense?

Lenat: No! No, no, no! Okay, take this away from this meeting: MCC had no connection to the US government other than being enabled by this law that passed and the fact that we were able to snag Admiral Bobby Inman, who basically retired to come and be its CEO. He

really was, by far, one of the smartest people, maybe the smartest person, I've ever met. The only person I've ever met who I believe had an eidetic memory.

David Hemmendinger: As a $40- or $50-million a year entity, what were the projects that the money went into? Was that Cyc?

Lenat: There were deductive database projects that led to what I would call modern databases. There was some various successful, up until a certain point in time, 3D crystallographic storage of data. Up until the time when discs got really cheap, they would have been a really good idea, but turned out to be completely cost-ineffective, so that just died. There were human interface type efforts: a sophisticated user interface project that formed, things that formed, things like that. There were a series of projects, some of which have spun off to other companies by now.

Grad: But nothing else in the expert systems field?

Lenat: The AI spinoff that MCC has that's still around is ours.

Grad: CyCorp?

Lenat: Yes.

END OF THE INTERVIEW

