AI: Expert Systems Pioneer Meeting
Session 6: Expert Systems Applications in Larger Corporations

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
Denny Brown	Teknowledge
Ed Feigenbaum	Intelligenetics/Intellicorp; Teknowledge
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
Alain Rappaport	Neuron Data
Herb Schorr	IBM/ISI
Reid Smith	Schlumberger
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
Paul McJones	CHM, Special projects
Len Shustek	CHM, Chairman

Ed LaHay	SI SIG, Meeting Project Manager

Burton Grad: I don't know about you guys, but I found this morning fascinating. I'm not denigrating anything you guys said yesterday, but these took me in very different directions than I was thinking when we were starting and discussing other companies. It seems to be a break point there somewhere in that early 1980 timeframe, 1982 to 1983, some new directions. I think that's a fascinating part of the story. Look, as all of you know who've looked at the program, we have no idea where we are. I don't give a damn, to be very honest about it. We're collecting information. If there are things that need to be followed up on and gathered more, we'll do it through oral histories or through follow-on things with David Brock. This is a swan song for me though. Pioneer meetings take too much energy.

Now we're going to continue with our stories and our plan for the rest of the day. We will continue with the stories on the other companies. We're going to talk about Fritz Kunze with Franz Inc. We'll get Reid Smith to talk about what Schlumberger was doing. We're also going to talk with Herb Schorr about what IBM did. Those are different kinds of stories. They're more about the usage and applications.

We're going to talk a little bit about applications; we no longer have the whole morning, but we may have an hour. What we're looking for are things that were different, unusual, that broke new ground. That's what we'd like to talk about. If you think about something that you think would be appropriate for that, that would be nice.

The afternoon is going to be a little bit an odd thing. We are going to talk about why most of the companies ended and, yet, expert systems blew up. Everybody and their dog knows expert systems. The companies, in general, didn't survive. We'll talk that through.

At the very end, and this will be a relatively brief session, what took its place? What happened during the 1990s? What was that transition? Not just these companies end in many cases, but what was starting and why was it now a different thing? The story of AI and how it's used in the 2000s is a future story. That's not what we're covering. But a little bit about what was happening, whether the Kurzweil thing was a break, whether these were the changes, what was happening? Again, we won't get that in-depth, but we'll get some of that covered.

Franz, Inc.

Grad: With that in mind, Fritz, you're on. Talk to us about Franz.

Fritz Kunze: Okay, I need to set the context for this talk. Last night I woke up with a nightmare. I had been invited to a party, and I told the truth at the party. At the end of the party, everybody hated me.

I'm going to tell some true stories, but I told these stories in an OPPSLA 2008 talk entitled "Careening through LISP Minefields." I meant to make that public, but I had images I'd taken from a book that contained a bunch of monkeys. I couldn't get permission from the book author, but it should be definitely part of the historical record.

Let me set the context. I was a graduate student in Berkeley in mathematics. I don't like to admit this, but everybody around me was an Asperger's victim. I was probably no different, even though I may look different now, but I was really no different at all than the rest of these guys. When I was in graduate school, I started a company called Campus Computer Dating, because none of the other graduate students knew how to date women. They knew that I had a girlfriend, and they wanted one too. So, I got a lot of math graduate students married to art history majors.

I fell I love with a program called Macsyma, which was founded on something called Project Mac at MIT [Massachusetts Institute of Technology]. I thought this was really the coolest thing and wanted to sell this thing. I'd done a little bit of research and concluded that it had been funded by the US government. Remember, I was a math grad; from my naïve point of view, "This should be open source since taxpayers have paid for it," right?

So, I decided to start a company around this. In the process of starting the company, I discovered that a company named Symbolics had managed somehow to get a license for the Symbolics code. They claimed ownership to it and actually had done a deal with Berkeley to get the code from a guy named Fateman, who was a professor there.

So, that code was kind of locked up. When I was trying to figure out how to get the code unlocked, I decided to sell something called LISP. Now you guys all know what LISP is. Because Kurt told me I'm talking to the camera, I've got to tell you: LISP sounds like a speech impediment. It's a programming language, and it's the second-oldest programming language known to man, having been invented after Fortran.

Herb Schorr has a very scintillating joke. Would you just repeat it right now?

Herb Schorr: You'll kill me.

Kunze: Go ahead. Do it.

Grad: You're allowed to say it. Let's hear it.

Schorr: I'm allowed to say it? Fortran is a language invented by geniuses for use by idiots. Now you can figure out where they heard that.

Kunze: That's the first part of the saying. LISP was a language invented by idiots to be used by geniuses.

The backdrop was Ronald Reagan was president, and he had decided we needed to beat the Russians by outspending them and building smart weapons. Smart weapons for the Department of Defense and DARPA at the time was to build AI into weapons. They went to the contractors around them—you know, Lockheed, any of the larger defense contractors—and said, "You guys have to buy LISP workstations."

The only company at the time making LISP workstations was a company called Symbolics. The CEO was Russell Noftsker. He's still alive. You guys can get an oral history from him. Symbolics went out of business I think in 1987 with revenue of $200 million.

But, basically, they were selling large machines for $100,000. A lot of the companies in here were trying to sell software on top of these machines, and the price point was just too high. Neuron Data was clearly going to kill them all.

So, each one of these companies beat a path to my door. My door was actually a house in the Montclair part of Oakland. I get a knock on the door, and there's a Japanese guy from a large trading company called ITOCHU. He says, "We're looking for the headquarters of Franz Incorporated." I said, "Welcome to the worldwide headquarters." The Japanese government had decided to invest heavily in Prolog, but the Japanese people didn't really trust the Japanese government. They figured if Americans were doing something with LISP, they should have a backup strategy with LISP too. These guys wanted to get LISP to run on some workstation.

Franz LISP

Grad: Stop a minute. What had you built? Did you do anything? What did you have at that point in time?

Kunze: I had a public domain software programming language called Franz LISP. People say, "How did you get Franz LISP?" It's a marketing trick. Everyone said, "Oh, I've heard of that." No, they never heard of it. They heard of Franz Liszt.

They couldn't get it out of their head once they heard it. So, it's a marketing trick. Remember, I was a mathematician, okay? We're not known about business.

Grad: Had you built anything at that point in time?

Kunze: We had built a version of Macsyma. We were running Franz LISP on everything. Almost everybody in the world was using Franz LISP. It was the most widely distributed LISP out there.

Grad: That's what I'm trying to get back to. You had to have gotten there from some place. When did you start the company? What was your financing? Start there.

Kunze: I started the company in 1984. I brought in a professor, which was a mistake. I shouldn't be putting this on tape, I guess. We each put in $500.

Grad: Keep going. What did you do?

Kunze: Well, the first thing we did is we had a board meeting among the founders. They tried to figure out how to price the damn thing. One of the founders said, "We give it away for free." I said, "Why? We're running a business." He said, "Because it's the right thing to do." I got rid of him very soon after that. I went to some AI show, and I made a fake press release. I put it on tables throughout the trade show, and the next day I got people coming and saying, "We want to buy this thing right now."

Grad: Was it a machine you were selling?

Kunze: I was selling software. Pure software. You could get it for free if you got Berkeley Unix.

Grad: Thank you.

Kunze: So, Franz LISP was based on something called Maclisp, which was used to build Macsyma. Fateman had some money to make Macsyma work, but the VAXs had just come out. They had problems with the large virtual address space, but there was no OS [operating system] to run it. So, Fateman gave Bill Joy, who went on to found Sun Microsystem, some money to build a paging version of Unix so that he could run this stuff there.

That's all we had, and we had companies that would come to my door and say, "We need to buy a LISP to run on our machine." The one that comes to mind is Tektronix. They were building some 68,000-base workstation. "We really need this. How much would you charge us?" I had no idea, so I said, "I don't know. $100,000."

Why is that? My quick mind at the time was… It's broken now because I had a stroke. I told you guys that. I have a brain injury, so if I make any mistakes, the fault is because I had a brain injury. Sorry about that.

Anyway, for $100,0000 they'd give us money and we'd port a version of this LISP over. Now we'd go to venture capitalists and they'd say, "Who are you guys?" I'd tell them just what I'm telling you guys. They'd look at me and sigh, "So sorry."

There was another group at Stanford named Lucid. Many of you guys know Lucid, and I decided to try to make buddies with them. Went down and talked to Dick Gabriel. We'd go for long walks in Oakland and Berkeley, and I'd tell them, "We ought to join forces." He'd say, "No way." He didn't tell me until later that the reason they said, "No way," was because of this bias I mentioned. He said, "Frankly, we're better than you guys. We don't need you."

He didn't say that directly, but that's what came out a couple weeks later. So, he got a bunch of money. He raised like $5 million and got a bunch of customers. We raised no money at all. The only thing we had was my mouth.

Grad: You were apparently a helluva salesman.

Kunze: I had one guy say to me, "Do you know what you're doing?" I said, "No, I don't know what you're doing." He said, "You tell me stories that are just so great. I got to get more of it."

Funding from SRI

Kunze: I don't know how it happened, but basically, we didn't get any outside funding. That's not quite true. SRI came to us and said, "We have a venture capital backlog. We're getting all this money from the government to build LISP-based software. We'd like to commercialize it. Would you commercialize it for us?" I said, "Sure, if you give us enough money." He said he would force their venture capital fund to give us some money.

They put in $1.5 million, but the problem was this was probably 1985—I don't remember the exact date—and the software that they claimed to own, they didn't actually have ownership to. They had hired employees who had carried the software out the door with them. They hadn't done the due diligence. I ended up with this pile of software that nobody had clear ownership to. So, I sat on their $1.5 million and earned $150,000 a year on it and eventually brought it back from the venture capital for 25 cents on the dollar.

Common LISP Standard

Thomas Haigh: Wait, wait, wait. Did you say you were giving the software away with BSD?

Kunze: Franz LISP was given away, but you have to understand the government. Sorry, that's a good question. Let me make sure you guys understand. The government put all the LISP vendors in a room together—Endless D and Symbolics—and said, "We think LISP is so important for the future of smart weapons. We have to make a new standard for LISP." That was created and called Common LISP. So, yes, the software Franz LISP was being given away for free. Who would buy our stuff? We were telling people we were building a Common LISP. We had no choice.

Unidentified Speaker: Also, for the record, who is the government? DARPA?

Kunze: DARPA. Yes.

We didn't need venture capital, because companies would come to us and say, "We need your LISP." And I'd say, "I need a machine," because machines were expensive back then. You know? They'd give me a machine, and they'd say, "Can you buy this?" I'd say, "No, what do you think we are? Rich? I don't have any money. You have to give me the machine, and I want you to give me maintenance for the next year." They'd grumble about that, but eventually they would say, "Okay."

Grad: Let me follow that point. Were you charging for the product?

Kunze: We were charging for something called Common LISP, which didn't exist from anybody at the time. We were converting Franz LISP to Common LISP, and our chief competitor was also using Franz LISP to make Common LISP.

Brian McCune: Lucid.

Grad: It was Lucid. What were you charging?

Kunze: Some company would come to us and say, "Give us LISP on our machine, so we can sell it to a defense contractor." We'd say, "What do you run on the machine?" If they ran a version of Unix, we would sell them a version of Franz LISP just before it became Common LISP.

Grad: I see.

Kunze: It was difficult to convert Franz LISP to Common LISP. We had to rewrite it from scratch, basically.

Grad: Yes, but were you asking $100,000? Is that the number you were using?

Kunze: I was using whatever I needed to meet payroll. If $100,000 was this month, you know. And it worked.

Grad: Was there any follow-on money that you were going to get from them?

Kunze: Well, the idea is that they would buy the product, and they would ship their machines out. They would buy licenses from us.

Grad: Oh, okay. Somewhere there was a model that made some money. Okay. Does that answer your question, Tom?

Thomas Haigh: Yes. I was just wondering. What did the venture capitalists think they were buying a piece of if you were giving the software away? The answer is the venture capitalists thought they were buying a piece of what they hoped would become a large market for Common LISP. Is that correct?

Kunze: Yes.

Lucid and DARPA Defense Contractors

Grad: Okay, continue. Now what happens?

Kunze: We made our way fighting this company called Lucid and, most of you probably even know this, this guy named Dick Gabriel from Stanford. Then we started running into a problem. We had OEMs [original equipment manufacturers] coming to us and saying, "We want to buy your Common LISP. You're easier to deal with than Lucid. But the problem is we were told by people in DARPA that if we buy anything other than Lucid Common LISP nobody will ever buy from us." You know, that's the defense contractor.

I told this to Brian, but I don't want to give any names. I'm not a Trump supporter, but he described Washington as a swamp. I will tell you why I agree with him: One of the people inside of DARPA was the nephew of the lead venture capitalist in Lucid. They decided to tell all the hardware manufacturers that they had to buy from Lucid; otherwise no one would buy from them. I discovered this, got really pissed off, and sent DARPA a letter that said, "You know this is going on. You apparently didn't do your due diligence on your program managers. They have to show if they have any financial effect." And DARPA did nothing. They hid the problem.

Even to this day, I know the name of the guy. Brian asked me last night when I was drinking," What was the name of the of the guy?" I told him and his face fell. Maybe that's why I had the nightmare last night.

David Grier: What time frame is this? Because DARPA has a very public rule that it adopted. It's named after this undertaker from Indiana.

Kunze: It's 1985.

Grier: The story that they tell is there's this undertaker who got stuffed, because there was an operator who sent things over and they said, "Everything has to be public. It's this rule." It's a common thing in DARPA lore.

Kunze: Well, all I know is the information that I'm telling you about. One of the founders of Lucid was a guy named Rodney Brooks. He went out and formed the company iRobot. He was in Osaka with me at a conference, trying to drink me under the table. He was an Australian who loved to drink. He blacked out and he told me the whole story just before I blacked out. I was young enough that my short-term memory got converted into long-term memory.

Doug Lenat: I don't want to put you on the spot by asking you who it was, but, Brian, who was that?

McCune: I'll tell you just before you black out.

Kunze: Brian rightly believes that destroying this poor chap's reputation is not worth it. Personally, I don't care.

Lucid Bankruptcy

Grad: What happened with Franz?

Kunze: Eventually 1987 rolled around and Lucid, our chief competitor, had raised $15 million. We had raised zero, except for $1.5 million from this failed SRI deal. Lucid was going bankrupt, and they came to us and said, "We'd like you to make an offer to buy us." So, I made an offer to buy them, which they accepted.

Now, there's a lot of subtleties around bankruptcy, but basically in bankruptcy, anybody can make an offer. Lucid had a big deal from NTT Data around their C++ system, which was being built to sort of replace the expert system companies in the world. I realized that my bid was not going to win, because it was so little money for the Lucid, because it was a liability. I think I offered them less than $1 million or something ridiculous. I withdrew because NTT could pay infinite money compared to me at the time.

So, Lucid went bankrupt. We sort of won, but what did we win? All the expert system companies were on the ropes. All the companies in here were probably customers of ours at one time or

another. And all the drama I heard yesterday, I knew what was going on, but I didn't really pay attention, because I was trying to stay alive the whole time.

I have this very funny talk called "Careening through LISP Minefields." It has a whole bunch of data, which would be very useful to the museum and you guys would probably like to hear it, but Burt would kill me because he said, "No slideshows." I may put it on the web, but I've got a big problem: You guys can look at it, but you can't give it to anybody, because it's got these pictures in it.

Grad: Keep going.

Change in Franz Leadership

Kunze: What happened to the company? Eventually, I got tired of fighting with my founders, so I left the company in 2007. The company is still alive. It's selling AllegroGraph, which is selling LISP and selling a triple store of Graph Databases.

I then started a company that is selling artificially intelligent robots now, and I've shown some of you these creatures that you talk to.

Grad: Did they buy out your stock? How did you leave?

Kunze: I left as a major shareholder. Basically, I am some sort of adviser to them. Every once in a while, I go out to lunch with the CEO, who in fact I hired, and I give him advice. They never follow any of my advice. I've concluded young people are basically untrainable.

Grad: You have not been involved in the company for the last 10 years or more?

Kunze: You know, I give them the advice, which they ignore.

Grad: But you still have ownership?

Kunze: I still have a large share.

AllegroGraph and Triple Stores

Grad: Any other specific questions? Doug, go ahead.

Lenat: Just as a capstone for that, we are an active, large customer of Allegro Common LISP, or at least we were until this talk. My confidence level has gone down a little bit.

Denny Brown: Yes, I think this is an important thing to note. We are talking about legacy here. AllegroGraph is, you know, essentially the triple store that's built to scale.

Lenat: We don't use AllegroGraph.

Brown: Okay, but in any case, their main product is an example of a type of database that is gaining greater currency now within enterprises as concepts of things like knowledge graphs. I know Doug has a particular take on the way in which that's being positioned within the industry now. But there's a lot of uptake right now either in the commercial, scalable triple stores, or Graph stores.

Grad: What's a triple store?

Lenat: You better explain that.

Brown: Sure. It's a database that is designed to make it very easy to retrieve very, very simple facts, as Doug explained earlier—essentially, subject-property-object tuples—and make it easy to be able to express something that reduces down to a declarative, logical description.

Grad: The true meaning of "declarative".

Brown: Correct. It lacks a lot of this kind of thing that Doug talked about earlier in terms of power of higher-order logic, but it's a very simple way of storing information that has become a path through the efforts of people involved in, for example, semantic web efforts to integrate data across things.

Grad: When we get to this last session that will be one of the "What's remaining?" things. What's the residue? What are the things that are still being used?

Brown: This is an example of a thread that has persisted and now has some signs of showing greater currency and adoption beyond its original ordinance.

Grad: Anything else before we go to the next one?

Kunze: I should remark that the company is totally dependent on LISP. It's going to be working on LISP forever. And the company I started, which is called Pandora Bot—another pun, forgive me—is also built entirely on a very large LISP-based system.

IBM Expert Systems Applications

Grad: The last two are different business models and usage of expert systems. One is Schlumberger and how they used expert systems to add on to the services that they were providing to their client base. Herb Schorr has a study of a different one: IBM, I gather, was at some point considering producing or marketing some kind of AI product. Herb was tasked, as he will tell us, with being the evangelist, I guess, is one word for it, for the use of expert systems in IBM. If I have said it wrong, Herb, you will correct me, I'm sure.

Schorr: Well, they didn't have an evangelist. My last job there was of bringing IBM into new businesses. Video disc technology, things of that sort, and artificial intelligence. I had started that in Yorktown and so on.

There's not much to say. I was given charge of development, marketing, and finance, but I didn't complete much of this because after about two years I left, so I really can't tell you much of the story or what happened. We went out, and we talked to a lot of customers. That was IBM's thing. It was very good. I managed to get a trip on an SST, a supersonic jet to London. It was Barclays or Lloyds Bank and the IBM salespeople wanted me to present what we were doing. I said, "Well, I have to go to the corporate management committee in the morning and you wanted me there the next day. The only way to go there is with a supersonic jet." So, I got that trip. The marketing people spent more money… I had to go back with a normal plane.

Nevertheless, I found a good deal of interest in talking to chief executives. I don't know what to say. Basically, I was much more impressed with the European CEOs than with American CEOs, not least in their grasp of technology.

Grad: Help me here in a minute. According to McCorduck's book, there's a bunch of stories about the applications, the expert systems applications, that IBM did within the company that spread very widely. I thought you were involved in that.

Schorr: I was.

Grad: Tell us about that a little bit. What were you doing?

Schorr: I didn't realize you want me to talk about this. You know, you're talking 30 years ago.

Grad: Do you remember any specific ones that you were involved with or that you got going in the company?

Schorr: Well, we did some configurators and things of that sort. Ed, I think, captured this at one point. We demonstrated expert systems to the corporate management committee. We

were doing a real, live demonstration showing how you could change rules in an expert system and so on. The boardroom had been set up so that you could do these demonstrations.

I found out afterward, in all the years of Armonk [New York], no one had ever done this. This terrified the hell out of the data-processing center. They dumped everyone off the computer for a few hours before so everything would be good, so there would be no chance of something bringing it down.

It made the point; we changed the rules in real-time, and they liked the power of the technology. We talked to some of the companies around this table. We wound up with IntelliCorp and we made an agreement with them. I found out about lawyers. The CEO of IntelliCorp and I wrote a two-page memo. The lawyers in Armonk took one paragraph and made 20 pages out of it. The CEO and I looked at it and said, “Well, if we don’t want to make this work, all this paper is worthless. But, nevertheless, let them have fun.”

Grad: Is that what then motivated IBM to then looked to see if they could do products in that area?

Schorr: Sure. We were susceptible to the same hype as everyone else. They were looking for new areas to go, and I had done this more or less for all my life, within research. I was on the first RISC project for IBM and so on. So, they put me in business.

IBM and IntelliCorp

Grad: Nothing was ever announced as a product by IBM. Is that correct?

Schorr: Not during my time. I left, as I said.

Grad: Does anybody know if anything ever announced afterward?

Ed Feigenbaum: Yes, there was an expert systems show. I don’t remember the name of it, but I know it was popular and it was good.

Grad: So, they did go into it.

Schorr: It was licensed from IntelliCorp, no? I don’t think we developed it ourselves.

Grad: Do any IntelliCorp people know? The question is this: Apparently, there was some kind of contract between IntelliCorp and IBM that they had the rights to license or use your system.

Friedland: Well, yes, that's true.

Grad: Did that continue? Did IBM release something using your software?

Friedland: I actually don't know. I know that we did some joint demos in marketing at the trade shows like AAAI. But I think it continued past when I was actively involved, which was 1987. I'm really not sure.

Grad: Up until now you've answered every question we asked. You must be a good consultant: You have an answer for every question. It might be right or not, but you have an answer for every question.

Friedland: Well, it definitively ended in 1988 when Herb canceled it in a fit of pique

Schorr: Me? No, I didn't cancel it.

Friedland: I know that there was an arrangement. I don't know how extensive it got or how long it lasted.

Grad: According to the book, *Rise of the Expert Systems Companies*, there is an incredible number of stories by large corporations of what they did and what the benefits were. Certainly, in there at that point, IBM's usage was very extensive and very profitable inside the company.

Haigh: In terms of IBM, it was an enormous thing, and it supported academic-style research labs in a way that companies don't so much anymore. I was wondering in terms of that IBM model; do you think there was any effectiveness in terms of lining up the work that people were doing on the research side with what was happening and used internally with the company and what made its way up to customers?

Schorr: I don't think Yorktown really developed any shells by itself. We were working on Prolog thanks to the Fifth Generation Project. I don't know if we could buy that or not, but we had to look at that. My memory is fuzzy about all that, but I think all these guys had product. There was no sense buying it and making it ourselves at that point. We knew less of it.

Paul Harmon: For historians, the newsletters that I filed are 10 years, written month by month. There are, at least, three or four issues that are devoted totally to IBM. Now, can I remember the details? I am no better than Herb, but if you grab me and I find the issue, it'll list any tools that IBM was pushing at the time. Right? It will certainly list all the applications. By the same token, there's issues devoted to DEC, and to the other major vendors.

They were big shows, and IBM played a major role in several AAAI shows. Each time they had a booth, and each time there'd be handouts on 25 applications that IBM was promoting, because IBM had done them or IntelliCorp had done them and IBM was using them internally. Later on, of course, AI Corp and Aion became the big leaders in mainframe.

Grad: They were using IBM mainframes?

Harmon: Yes.

Schorr: Oh, yes, we talked to Reinstein, I remember that. Then I did, in fact, a keynote where I found out how incompetent the press is. They are in the swamp with everyone else. After the talk, I got interviews set up for the rest of the day and no one had bothered to go who listened to the talk. First question out of every newspaper person's mouth was, "Please, summarize what you said for us." By five o'clock, I was groggy.

Expert Systems and the Seven Dwarfs

Feigenbaum: So, downstairs in the museum exhibit, there is an exhibit that uses the expression that people around the table with gray hair know about: IBM and the Seven Dwarfs. I was on the board of one of the seven dwarfs, which was Sperry, which traces its history back to the ENIAC actually, I think. Or the UNIVAC. In all this hullabaloo about Fifth Generation, Sperry decided to get involved in the field of symbolic processing. But they were a hardware company, so they made a deal with Texas Instruments. They were selling TI's LISP machine, which had this custom chip that DARPA had funded. But they were selling it under their own brand name. It was a Sperry box.

As an observer, it didn't fly. I mean, it didn't do very well. As an observer, I'll tell you why I think it didn't do very well, and I think this is important. The experiment lasted over a period of about three years. The salespeople of Sperry were simply unable or unwilling to make the change from ordinary computation to symbolic computation. They couldn't sell and Sperry had no way of selling this thing, because the salespeople just were living on a different planet. Finally, Sperry just gave up on it.

Grad: Let me go back to my general question. DEC, Sun, all these others, their equipment was being used to do these functions. Did any of them go into the business of trying to sell something? To sell their own shell, to sell their own whatever product?

Harmon: TI certainly did. TI built a LISP chip, but TI also had a PC version of a software tool for building expert systems.

Grad: Who else?

McCune: Didn't they have something from Teknowlege, or was that stolen? I think the story was they got a copy of E-MYCIN and sold it as is.

Harmon: Okay.

Grad: Any other hardware companies? I'm thinking of the seven dwarfs, the other ones besides Sperry. Okay, you answered my question. The answer was not that you're aware of.

Schorr: Well, I can't believe that DEC didn't have something.

Harmon: DEC had something.

Grad: Did they have a product? A service?

Schorr: DEC did configurators.

Grad: That's internal. I'm talking about external use.

Rappaport: They were a distributor of our product. They were a reseller of our product.

Grad: They didn't have their own product though.

Rappaport: Right.

Grad: Interesting. Okay. Peter?

Friedland: Oh, your question about IntelliCorp and IBM prompted me. Since I didn't have access to Cyc, I actually went to the web. I discovered something fascinating: Apparently, IntelliCorp is still alive, unbeknownst to me and Ed, I believe. It's alive as a partner building "intelligent frontends for SAP." It's still the same IntelliCorp.

Apparently, somebody kept it alive because I'm just reading the history and it's the same IntelliCorp. Remember, after KEY came a system called Kappa, which was just a renaming of the evolution of the KEY system. So, it's the same IntelliCorp, now building smart frontends for SAP lifecycle management. They announced the partnership with IBM in 2007. I guess we're still around.

Grad: Nothing ever dies. It just gets a new name sometimes. Okay, stop on this.

Schlumberger and Dipmeter Advisor

Grad: Next subject. You're up, Reid. Tell us about Schlumberger's marketing and use of this?

Smith: Sure. So, first of all, Schlumberger is the world's largest oilfield services company. Their revenue, even in the downtimes, are roughly $30 billion. Their R&D budget is something on the order of a billion dollars. So, big oilfield services business. Basically, anytime you hear BP is drilling this or doing that, it's really companies like Schlumberger who are doing the work. That was part of the business.

Peter alluded to another part of the business; there was the oilfield services business. That's where all the money was, and then there was the rest. But the rest was a couple-billion-dollar business for the Macs. That was in metering; it was in CAD/CAM automated test equipment. If you had an Intel chip in your computer, that was tested on Schlumberger equipment. They were the world's largest smartcard manufacturer and so on and so forth.

You all know better than I, as my time for this goes back to 1981. I've got an extensive bibliography on everything written on AI in Schlumberger in multiple centers, because I managed several of them over the years, which in the earliest dates I can find are late 1970s, 1979-ish. There may have been something earlier with Clarone I don't know.

In what I saw, in the oilfield part of the business in the day, there were two main threads. One of them, think of it as signal understanding. You're looking at data typically near both of those and trying to understand simple questions like, "Where do I perforate to find oil?"

The second thread was really program synthesis. Let me tell you a little bit about what happened. The part of the signal understanding work—think a.k.a. knowledge-based systems or expert systems in the day—the most famous one was called the Dipmeter Advisor, which has to do with interpreting literally the tilt of rocks around the borehole and extrapolating what the geology is far away from the borehole. You know, that's a fault, that's a distributary fan, or whatever.

That got commercialized. I think it was 1983-ish. It was commercialized in the following sense: That was used by Schlumberger to deliver Dipmeter services data to customers. When you asked me yesterday, "Was this products and services?" Everything Schlumberger sells is a service. All of it. Hence, oilfield services. What they would do is give you the data and, in the case of the Dipmeter Advisor, that was AI in support of improving on the interpretation of the data that was delivered to oil and gas operators.

Much to be said there. A lot of it has been recorded in various lessons-learned articles and so on over the years. I'd be happy to show you where that is, if needed. So, that's one thread. That thread continues to today in a number of ways.

If you go and look at Schlumberger's key software platforms today in oilfield services, one of them is called Petrel. AI technology of the statistically-based, neural net-based, and symbolic-based is throughout that software and used for a variety of applications around what's called "reservoir characterization," part of which has to do with understanding things near to boreholes and part of which has to do with understanding things better far away on a seismic scale. It never went away. It was embedded into what Schlumberger sells over the years and has continued to sell.

I was kind of fascinated yesterday. We went back to the symbolic or statistical, or symbolic versus neural nets kind of thing, and all I would say is if you're building signal understanding systems, my experience is, of course, you have to have both of those things. The rules operate on a set of features, where a feature is a fairly high-level construct and somebody has to turn the actual low-level signals into something that those rules can deal with. The one line of learning with the Dipmeter Advisor is the biggest impact we ever had on the performance of the system was to make a step change improvement in the feature extraction used down underneath the rules. That was the thing that made a huge difference in its performance.

Friedland: I'm just curious, for the Dipmeter Advisor itself, did you or any of the people below or above you ever make a monetary investment for this huge company, this multibillion-dollar company? I'm just curious, because if we attempt to monetize the value of expert systems, it's quite likely yours is maybe the greatest of all. So, I'm just curious. Did you ever try to do that?

Smith: No. I mean, this is a long time ago. I can't tell you even what the revenue was from various logging services. It's probably possible for someone to go and reverse engineer that, but that's not the kind of thing that I had in my hands.

Grad: Reid, follow that. Was there a separate charge for the use of that AI capability?

Smith: Yes, for a while, that was true. Ed will remember the number. It was $0.50 a foot.

Ed: Yes.

Smith: Literally. AI sold by the foot. You know why? Because these things are in boreholes, and if you are logging 4,000 feet of borehole, that's the way Schlumberger charged for all of its services in the day.

Friedland: You asked for $2,000 for a 4,000-foot borehole.

Smith: Absolutely. Some of that was delivered directly to customers. In fact, in the Computer History Museum in Boston, there once was a Dipmeter Advisor log. I don't know what happened to it. I've got lots of this kind of stuff, if you're ever interested.

Capturing Expert Knowledge

Lenat: Real quick historical notes. I was one of the people who worked on the Dipmeter Adviser with David Barstow. It's probably worth mentioning that one of Schlumberger's main motivations for doing that application was that their best expert, whose name I believe was Gilbreath, was retiring.

Smith: Al Gilbreath.

Lenat: They were scared to death that none of the junior people were going to be doing nearly as good a job as he was doing. If we could even capture a fraction of his differential expertise, then that would be highly cost-effective.

Smith: Yes. I worked with Al for quite some time. Al was a wild man. I mean, he was a man among men and a prince of a guy. But he'd drink you guys under the table; that'd be before breakfast, my friend.

Friedland: That's the first time that got mentioned. That's, I think, an important thing to capture historically, that in many cases, the most successful expert systems that all of our companies built was when a partner company was scared to death about someone retiring and losing their expertise.

For IntelliCorp, 3M had one guy who was their super expert on adhesives. Like, if they needed something in a hospital ER to glue a wound, this guy just knew everything about that. IntelliCorp worked with him to capture his expertise. I forget if it was IntelliCorp or Teknowledge that did a Campbell's Soup advisor system. Same thing. Was that Teknowledge?

Brown: At Tek we did a consumer consumable application, which we're still not supposed to talk about. But it was not for Campbell's Soup.

Friedland: Okay.

Smith: There's a double-edged sword here, because, of course, this particular guy was an oil finder, in the parlance. However he did it, he was an oil finder. He was very successful in a particular area of the world. It was the Gulf Coast of the US that he knew the most. Turns out,

it's very narrow; that particular expertise translates to various geological analogues. Nigeria is one of them in the Niger Delta, but it doesn't transfer everywhere. This doesn't make any sense in Canada, where it's a reef environment in the western part of Canada. Anyway, that was kind of interesting.

Schlumberger Software and Services

Smith: There's a lot to be said about all of the knowledge acquisition and generality. That's been covered in a number of areas by a number of people. But let me tell you about a different thread entirely. A different thread had to do with program synthesis. So, we had two of Brian's automatic programming colleagues working at Schlumberger: Elaine Kant and David Barstow.

McCune: Dick Gabriel, by the way, was another one.

Smith: The work Elaine Kant and David Barstow did went in a number of different directions, but one of them was this: they built the software architecture that was used in Schlumberger's field units. Every service that Schlumberger delivered was based on a software architecture that those folks had devised based on what they had learned working with Cordell. That was kind of an interesting thread. It's not a product that anybody ever saw, but everybody saw every time the service was delivered. It was just invisible to them.

It also led in a different direction. How many of you ever had experience with connection machines? You remember? I know they were kind of thrust in the world of massively parallel machines. Well, Schlumberger first had a CM2 and then later a CM5. What we did with that was two kinds of things. One kind of thing was essentially writing code to simulate electromagnetic and acoustic wave propagation in 3D. That was one kind of work.

There's a little side story, which I will tell you in a minute, but the second was what's called 3D depth migration. Basically, all seismic signals, all the reflections are in the time domain, and of course to be useful, you've got to translate that into the depth domain, which is not as easy as it sounds. We used CM5s to do that, and that business got to be quite profitable after they went out of business, because you could buy the nodes on the ultra-cheap. That turned out to matter in that business. That was another thrust.

The little story that goes with that is that Elaine Kant, another AAAI fellow, didn't like to deal with customers at all. She worked with me. I managed three centers in Schlumberger: one in Palo Alto, one in Austin, and one in Cambridge, UK. Elaine worked with me in Austin, and before that we had worked together in Ridgefield, Connecticut, as we were talking earlier. Anyway, you know what she did? She started a company doing the same kind of software synthesis, if you

like, and I thought, “Well, gee, you didn’t like talking to Schlumberger customers. So, you go start a company. How’s that working out for you in terms of who you get to talk to?”

Well, it worked out quite well. She went from simulation of acoustic wave propagation to pricing derivatives. That turned out to be the profitable niche early on. I don't know what it is now. You’d have to ask Elaine. She is still in Austin. I don't know if you ever see her.

Lenat: Yes, we get together about once a month.

Friedland: What was the advantage of a connection machine? Why did you use a connection machine for that?

Smith: It’s by its nature amenable to massive parallels. This is one of those “if I knew then” kind of stories. We went shopping for machines and we went and talked to now-chief of Oracle, who had bought a company, which I’ve forgotten the name of. They were selling parallel machines, and for our purposes, we were better served by that particular architecture than we were by a competitive architecture. So that’s what we did.

Let me end this story by talking a little bit about the non-oilfield part of Schlumberger, and one of the things that was done there and the work that was done by Harry Barrow and Glen Kramer. Harry Barrow is well known to, I imagine, a number of people in this room, though Glen perhaps less so. They worked together in the early days of what’s now called parametric design.

In the early days of CAD/CAM systems you couldn’t say things like, “This part has a hole in it and here’s the radius of the hole.” You know, what you had to do was translate that into a set of points. What they did was to build the first parametric design system I am aware of in support of Schlumberger’s CAD/CAM business. They could also simulate the behavior of the mechanisms that they were able to design. That to me was the one area in the non-oilfield part of the business where it was the most obvious to use things that can trace their roots to AI.

Other Schlumberger Applications

Smith: Let me just say one more thing. As many of you have alluded to, there are some things that have come out of this, what we would agree are sort of straight up AI or at least were regarded as AI in the day. But a great many other things happened that were done by AI people, but got applied in different areas.

VSAT would be an example. You know, these are the early days with Hughes and Schlumberger, who did satellite communications. They did it so they could deal with small antennas next to Schlumberger vehicles. Again, some of the people involved you may know and

who have continued to work on AI-related things to this day. But a lot of infrastructural things happened that were part and parcel of what you had to do.

Here's my little story. When I was a student, hubris was pretty common. I'd agree with that. It was so common that, in fact, this is the way one almost thought about the world. Here's the world of computing; this is the AI part. Then there's all this other stuff over here in the corner. You know, right? Does this sound familiar to anybody?

Of course, what happens when you try to build real systems and you field them is this is the world of the things you have to worry about, and here's the part that is the sort of the smart bit over here in the corner. In other words, you've got to get data in and out of these systems. They've got to integrate with other kinds of machines. They've got to run on hardware that people can actually support and transport around the world. That's where most of the time went.

With me, as a specific example is the Dipmeter Advisor, I wrote all the graphics system. To be honest, I think the graphic system had a bigger impact than anything! You know, those were the early days of what we would now call intelligent assistance, where it's a man-machine symbiotic relationship, and the machine is doing some things like consistency and scale and all of that stuff that people don't do that well. It frees them up to do the things that they do extremely well.

Grad: Thank you, Reid. Any specific questions before we get into our next topic?

Schlumberger and AI Engineers

Feigenbaum: Reid, there was a human engineer in Cyc, who died: Michel Houyer?

Smith: Yes.

Feigenbaum: After he died and you left, did any of that continue? Do they still do AI? Do they hire AI people?

Smith: You may remember, I started off in Ridgefield, Connecticut, running a group called Knowledge-Based Systems. I later moved to Palo Alto in 1985 and originally ran a group called Knowledge-Based Computer-Aided Engineering. Later, I was the director of what had become as a result of the movement out of Fairchild and the pulling of the people that used to work for Peter into the rest of Schlumberger. I ran that center, and AI was a big piece of that. Not all of it, but it was a big piece.

Then I later ran Schlumberger's Cambridge UK center. That was far away from AI, and in fact when I arrived, it was far away from computing. This is rock mechanics, fluid mechanics, drilling fluids; seismic was the closest you got. They continued to hire people with that background, but

they've started a new center, Ed, in fact, on Sand Hill Road, which is the Schlumberger STIC—Schlumberger Technology Innovation Center, something like that. So, they're doing it again.

Indeed, the company that Bruce Buchanan—well known to Ed and several others of you—and me and Eric Schoen are running these days is in the business information discovery. Basically, what we're trying to do is to help you find, select, and analyze data in unstructured documents (i.e., text). There's a lot of interest in that today. Companies everywhere are building data lakes banks, and there's a good deal of interest in interpreting the unstructured data in data banks as well as interpreting the structured data. So, that's continued and Schlumberger has made a return to at least Menlo Park.

Feigenbaum: Schlumberger and Menlo Park are using your system?

Smith: Yes. For us now, our biggest thing is the Society of Petroleum Engineers.

Feigenbaum: Do they hire AI people?

Smith: Yes, they do. They've just help put in place a VP of—it's probably called digitalization; I don't remember the exact title—but there's a fair amount of that. You'll find digital transformation groups. For example, some of what was content management turned into digital transformation, and they're doing that sort of thing.

I was talking to Eric Schoen, who retired, before this session. This is pretty good actually:

Eric was a student of Bruce Buchanan also. He did a lot of the networking work at SUMEX-AIM, as Denny will remember. Eric retired as Schlumberger's chief software architect. Well, it's not a bad job for a technically-oriented AI kind of guy in a company that is not dominated by that sort of software.

So, yes, they do continue and there are lots of specific algorithms that are used. It's sort of like it's gone into the fabric. You once said this to me about expert systems.

Evolution of Expert Systems

Smith: One of the things we have done in my little company, and you can see this in AI magazine and other places, is we analyzed all 30 years of the innovative applications at the AI conference. If you want to go look at what happened, that's a lens through which you can look. It's not all encompassing, but it's not bad over 30 years. Herb and Alain were the co-chairs at the outset.

Anyway, you can see the change: In the early days, it was all expert systems, all the time, 1989. Nowadays it's not. It went down and other things came up like machine learning, natural

language processing, vision, robotics, and so on. One thing Ed once said to me is it's not that the expert systems went away; it's just that they weren't innovative anymore. They kind of disappeared into the fabric. Do you still believe that?

I wondered yesterday. I tried this out on a few of you, and I had the impression when we were talking as if that all just went away. Gee, you know, it's gone. I'm not sure it's true.

Grad: You're going to another session. Do you have one final question?

Hart: I was going to ask Reid, how's the past affected the present? What exactly is going on at Schlumberger that had its roots in the stuff we've been talking about for two days?

Smith: I can't speak definitively to that. You know, I only see it through the narrow lens of what Bruce and Eric and I are doing with them now, which is interpreting data in text. So, I can't tell you what else is going on there. I just don't know. But Peter will remember this well. Sand Hill Road was far away from the mentality of Schlumberger. He and I had the "pleasure" of running centers that were not regarded to be central to the business of the company in Palo Alto. That appears to have changed with a changing of the guard and the management.

NASA Shuttle Refurbishing Scheduling System

Grad: We're now going to change topics, so put a different hat on. We're not going to talk about companies anymore. I'm trying to keep enough time for the rest of the topics. I'd like to talk about three major applications, which showed breakthroughs. I'd prefer to have one in the scientific-technical-medical field, one in a business field; and the third up for grabs. So, shift from listening to companies—why they succeeded, didn't succeed, what the differences were. They were fascinating stories this morning.

Here's my basic questions: Think of an example where you can tell me what the problem was. How did you find the expertise that you wanted to capture and the expert to talk to? How did you capture that information? How did you store it, hold it, keep it? How did you make sure you had the right logic and rules to work with? How did you "program" it or get it to be operational? And finally, how did it run on a regular basis? Think about something. and it's almost like I'm going to have no choice. Whoever puts their hand up first and says, "I got one!"

Think about something that you think was transformational that opened a door that was different from the ones that all the other good stuff that was being done. I'll give you a minute to think. How is that? I want to give everybody else a chance here, too. And how much money are you donating to the museum? Whoever gives the most money gets the first call.

Friedland: I like the sound of that.

Grad: Len told me to say that. Okay, which of you would like to tell a story? How about somebody, anybody, on the scientific-technical-medical side? Pick a story that was really transformational.

Friedland: The largest single payback we had from all the systems we built at NASA measured in a direct monetary gain was the system that Monte talked just a teensy bit about yesterday and would've talked about if he was talking about Red Pepper today. It was a scheduling system that incorporated the expertise of two experts.

Grad: What was the problem? Start with that.

Friedland: The space shuttles when they returned to Earth needed to be refurbished. That process takes anywhere between a month and two months, and it's everything from replacing the tiles to fixing problems with the reaction control system, refueling the various entities on it. That takes place in something called the orbiter processing facilities, the OPFs, at Kennedy Space Center [KSC].

It's a very complex scheduling problem, because there are a lot of complicated constraints. For example, when you're replacing the tiles, they use highly toxic chemical adhesives, so no one else can be in there. The people doing it wear protective garments and so on. It's a very expensive process; costs over a million dollars a day in labor time, materials time to refurbish the orbiters.

Prior to us getting involved, the process of scheduling was almost entirely human. There was not even computational assistance really to it. In fact, Monte used to tell me the budget per orbiter refurbishment for paper and X-Acto knives was $12,000 because, literally, they put long sheets of paper along the wall and a single human called a flow manager was in charge of that entire process.

The flow manager for the Endeavor shuttle was a guy by the name of Eric Clanton. Monte's system was sort of a combination of incorporating the human heuristics of Eric and another guy, Wayne somebody, who was in charge of Columbia, who also agreed to work with us, and a very interesting just-in-time, constraint-based scheduling system that is well documented in a book. Mark Fox and Monte wrote a book after this called *Intelligent Scheduling* that gives a lot of technical details about this system among several others.

When the system was implemented at KSC [Kennedy Space Center], it was very interesting, because it shows the transition time in computer science. I always have to lecture the Air Force people that, from very basic research to field, the application is often measured in months, not many, many years as it is, for instance, in high-energy physics.

Monte and several of his colleagues developed the research necessary to build this just-in-time scheduling and rescheduling system. It won several awards from publications, and then within another six months it was implemented fully at Kennedy. It eliminated the need for weekend overtime for shuttle processing, which saved a minimum of a couple million dollars per refurbishment. That's documented by the fact that NASA has a thing called Space Act Awards. It's unusual for a federal agency, because it can give actual monetary rewards to NASA employees, small monetary awards for improvements in science and other areas that are hard to measure monetarily, up to $100,000 or one percent of the total savings of the innovation to the team. I had the pleasure of applying for that award for the system, and it produced the largest Space Act Award ever given at NASA to this team.

Grad: How much?

Friedland: Well, it hit the max. It was $100,000 because the savings that NASA estimated was well over $10 million. It was sort of a combination of expertise, heuristics from those guys, with a very, very nice constraint-based scheduling system, which had the feature that you could stop at any time. It may not produce an optimal answer, but it always produced a correct answer for that problem.

Grad: One that would work.

Friedland: One that would work. It wasn't guaranteed to be the best, but it was guaranteed to be complete.

NASA System Tools and Hardware

Grad: So, tell me, were there any particular tools that was used in getting the knowledge and information from the people?

Friedland: It was hand-mediated by Monte and his team. You have to ask Monte and I don't remember exactly. I don't believe they used any specific tools. They hand-coded the heuristics to guide the scheduling system. I don't believe they used any particular knowledge of any tools.

Grad: Any particular inference engine?

Friedland: No, I mean, some of the work was done on TI Explorers, which thanks to the goodness of our partnership with DARPA we got at less than half the cost that Herb mentioned. We got them for less than $50,000, not $100,000, because DARPA had negotiated a big deal with Texas Instruments and allowed us to glom onto that.

Grad: Then you ran the thing on TIs?

Friedland: No. It was converted to run on the IBM hardware actually.

Grad: So, 360, 370s.

Friedland: It originally was a mainframe system, probably 370 down at KSC. Then KSC converted to Apollo workstations, and I believe was converted to run on those systems then.

Grad: Any questions on that application for Peter before we go into another one? Okay, thank you. It was a good story.

Ricoh Customer Service Application

Grad: Who else is volunteering? That's a sort of semi-business story, isn't it? Peter, are you next? Are you making the next big contribution to the museum?

Hart: Well, my wife and I wrote the museum's newsletter for many years. I thought that was a contribution.

Grad: Do we have a dollar equivalent for that?

Hart: Well, what I always told Len is you can't go wrong. It's two for the price of none.

Friedland: You mean, Peter, was the achievement worth it?

Hart: Every nickel.

Grad: Peter, go. Do you have a story?

Hart: I'd like to set the bar and the expectations extremely low for this contribution. But I'm going to skate it out because I think it has some technical aspects of interest that are unique and would also extend this thread of "What's been going on?"—not exactly today, but much more recently than the 1980s.

I alluded to this very briefly yesterday. Ricoh's a large copier-printer-office machine company. We have a national helpdesk like every other company. You call and they say, "Hello, this is Ricoh." And you say, "My copier's making streaks," or something like that. Like everybody else, they're pretty low paid, high turnover workers. You need to provide a lot of computer support, so they sound more effective when the customer calls in.

This was about a one and a half-person effort for a year or two. So, it was an extremely inexpensive development effort. We designed this system, which I mentioned yesterday, that we called query-free information retrieval. It was based on an expert system for diagnosis that was by now classic Bayesian net, as opposed to rule-based and an older format that would do the troubleshooting. That had a pretty good sensitivity analysis, so it knew what question to ask next.

It also had this, on the side: what was the person likely to ask and what information do you have to present to the helpdesk operator? We had just a little tickler on the screen that says, “If you want information about this exact thing, click here. We’ll pull the right manual off the shelf, open it up to the right page, and get to the right diagram. That’s what you should be talking to the customer about next.”

It wasn’t the biggest deal in the world, but this very low-effort thing was used for 10 years straight in our Chicago helpdesk office. I don’t know how much it saved; they never told anybody. They never told the members, but it was a very successful application made at very low cost.

Grad: Who was your expert or experts?

Hart: The people who understood how to fix copiers and printers and things like that. This is not rocket science; it’s just that you’ve got somebody calling up and says, “My copier is making streaks,” or “there’s blank pages. What shall I do?” The payoff is extremely high because it saves a trip by a guy in a truck to go out there. Then if you do have to send a guy in the truck, you can send the guy with the right obscure part, so he only makes one trip and not two trips. As I said, it’s a very kind of low end, simple application, but very cost effective and with some new technology.

Grad: Let’s go through our question list. Were there any specific tools you used in collecting that information and recording it?

Hart: No, but it was a very fun piece of software on which we built this. Probably everybody here knows Eric Horvitz or knows of him. Before Eric moved to Microsoft, he and my student and his good friend David Heckerman started an expert systems company. In fact, David won the ACM prize for the best dissertation of the year the year his thesis came out. They had a little Bayesian net tool, and I said, “I’ll be your first customer.”

So, we built our software, our expert system there, on their little Bayesian net thing. They must have had, I don't know, six man-months into it or something like that. I think we were their only customer ever. Then they both moved to Seattle, and we just kept things moving on.

Grad: Okay, that's how you captured it. In doing the rules or guidance or selection process, how did you know which manual, which page?

Hart: Yes, that was very straightforward engineering. You know which copier, you know what part of the system you're working on—all that was very elementary kinds of stuff as I mentioned. There was this sensitivity analysis thing that said, "Here's the next most cost-effective question to ask," because you want the operator to ask the fewest possible questions to isolate the fault. All that stuff. It was just really simple stuff and just extremely effective.

Grad: Was there particular hardware you were running on, a particular system software?

Hart: I think it was just a workstation at the time.

Grad: Sun, or things of that ilk?

Hart: No. I don't remember which one. Just a workstation.

Grad: Any questions on this relatively simple, unimportant application?

Hart: I set the expectations as low as I could.

Grad: And you did! You succeeded.

Expert Systems Morphing into Foundational Systems

Feigenbaum: My only comment is Xerox PARC AI people did the same thing for Xerox machines. Danny Barbrow died last year, but if he were alive, he'd come here and tell you all about that. Those Xerox field people are just sort of wedded to that system.

Grad: I am guessing that everybody who's doing field maintenance, customer service, everybody has done something of this ilk, is what I'm guessing. I don't know how you would run without that. Brad?

Allen: Yes, I think this is another example of an area where a significant amount of expert systems work ended up morphing into kind of the foundation of a certain class of application.

Grad: You've mentioned CRM [customer relationship management] before. Basically, this is an example of that kind of thing.

Hart: Yes, I think there's an interesting example, and I agree with what Brad said. Ed and Bruce and Ted started out with MYCIN saying, "We're helping doctors. Why don't we start with the most highly educated professionals in society?" It turned out to have this very widespread use at a relatively early day. You know, let's hire these not-much-more-than-minimum-wage workers, as I mentioned, with high turnover and help them. If you have enough of them, you create quite a bit of cost saving for a company.

Grad: May I ask you about an analogy here? There was work being done, 1960s, 1970s, on computer-based training of various kinds. I talk about the complexity of trying to pre-write what happens when the guy goes off the thing. I remember doing one for healthcare for Equitable Life when they'd do healthcare systems, and they wanted to check the things out and deny the things as quickly as they could. They were successful. We helped fight it. This, to me, is a far better solution. You still have the human in the thing, but you have really provided the information to guide them through the process. Is that a fair statement?

Hart: I think so.

Expert versus Search Solutions

Allen: What's interesting is I've heard a couple of times over the last two days this theme coming out where a lot of really practical applications today we would go back and talk about those things as search applications. Essentially, Verity, which was really kind of a foundational company in terms of enterprise search, was burst out of the work that Brian was doing at ADS. This is another example of something that if you gave that task to an engineer to kind of solve that problem today, they would basically treat that as a search engine task.

Grad: Tell me what the differences between a search-based and an expert systems-based solution.

Allen: Well, what I'm saying is that some of these applications effectively went in and built using expert systems technology. Today we would use a search engine to index that information. We give people the ability to retrieve that information in a way that's custom fit to helping them in their decision-making process.

Grad: In one case you have to ask the question in the right way. The other place they were guiding it though, weren't they?

Allen: Yes**.**

Hart: I think that the one thing you have to add to that is that most of these things you need some sort of probabilistic or uncertainty reasoning, because there's no perfection in this diagnostic process. You could somehow have a formalism that supports that.

Allen: Yes.

Smith: I like the allusion to search. Imagine your search engine knew your domain of discourse? Imagine it knew your company as well as what it knows about you. Today, those statements are not true. You know, your search engine does not know your industry. Your search engine does not know your company. One of the results of this is when you go talk to people about the number one application people complain about in companies, it's enterprise search.

McCune: It's because they took the added technology we put in the top of 1988 and people didn't use it. They had to develop their own knowledge bases, which turned out to be the search, and they refused to do that.

Smith: Today, then it's possible. Indeed, that's exactly what we do.

Friedland: Enterprise search became the backbone in the late 1990s. There was a flurry of importance of knowledge management companies. I started one: Introspect Software. There are companies like Open Text, which is still alive and sold many, many copies of their systems. And, of course, Microsoft's SharePoint. Those incorporate search engines in context for the company.

Turbo Tax and "AI-Based" Applications

Friedland: I do want to mention one thing. I had a question for Ed. Ed, you always used to say that the single greatest spread of expert systems was the fact that it was actually Turbo Tax. Do you still think that's true? Did that have the most influence of an existing expert system on the most people?

Feigenbaum: At the time we wrote the book *The Rise of the Expert Company* it was the most widely distributed expert system ever, because it was millions of copies of that rule-based system for system people in doing their taxes. I haven't tracked the innards of Turbo Tax to the year 2017 to know what's inside there now.

Friedland: I mean, they claim in their advertising "AI technology." Whether that means.

Feigenbaum: It's probably the same old rule-based system. That's my guess, because it doesn't lend itself to pattern recognition.

Harmon: Turbo Tax was certainly billed as an expert system. It was probably an expert system filled with flaws.

Allen: At the beginning.

Harmon: I interviewed the guy that built it, and he was excited about it and all that. What happened to the company and what's inside it today, I have no idea.

Grad: A separate question is: "AI-based" or the use of "AI," is that the equivalent to some of the other advertising slogans like "organic"?

Friedland: I was listening on the radio going home last night, and there's a real estate company that says, "We'll do the best job for you because we incorporate AI and machine learning for you." So, yes.

AI-Based Spirometer Application

Feigenbaum: I have a cute story to tell you that fits your list of questions exactly, but it's not one these big bang payoffs. I will yield the floor to Paul Harmon if he wants to tell us. He's the guy who saw all the systems. He wrote about them. He knows where the big bang is and so does Reid from looking at the innovative applications in AI proceedings. There must be some really, really big bang ones that you guys want to talk about. If you don't, I'm going to tell my little story.

Grad: Tell it. Go ahead.

Feigenbaum: Bill van Melle, one of our students and later a cofounder of Teknowledge, did his PhD thesis, the result of which was E-MYCIN, the shell, the first expert system shell, so that you could take the knowledge of blood infections out of MYCIN and you could stuff other rule-based knowledge into MYCIN and it would do something else. I decided I wanted to personally test that. This was now 1976. I had actually two connections up at Cal Pacific Medical Center in San Francisco. Both of them lead to expert systems. One was Larry Feigen's ventilator manager, but the other one was a pulmonary diagnosis system. My wife, Penny Nii, who was kind of a premier knowledge engineer, and I went up to San Francisco several times over a three-week period to work with one of their clinical pulmonologists, who would tell us what the interpretation was. Any of us who have lived long enough to have a pulmonary problem know that they stuff you in a box called a spirometer. and they seal up the box. Then you breathe in and out, and the volume of your breath in and out comes out in a traced curve. They can figure out what problems you have with lung disease.

How did we do it? That was one of your questions: How did we do it? Just the way we did with Carl Jurassic: start out in the first drawer of his 100 cases in a file drawer, start putting rules in. "How did you do this? How did you do this one? How did you do this one?"

Put it into E-MYCIN. Terrible, wrong answers, bad stuff. Case 50, not so bad. Case 100, still pretty lousy, but getting better. Okay, let's go to the second drawer. Start through the cases. It's getting a little better, it's getting a little better, it's getting a little better. By the end of this, 200 cases, it was pretty darn good.

We started on the third row, which is kind of the finale, and we were a little bit into the third drawer, not very much into it when our pulmonologist said, "You've got it all. These diagnoses that are coming out are just the right thing and we're wasting our time; we're at the point of diminishing returns."

So, we had all the rules. Turned out to be 400 rules, I think, for doing this. As so many people around the table have said, natural language is much simpler when you're inside a domain, because the specialists in the domain have their own natural language.

In my case, I call it med speak, or maybe pulmonary diagnosis speak. They have certain terms that they use, and we modeled that so that we had dialogue that would come out. The end result was a diagnosis you would read in several paragraphs with their pulmonary med speak. By the end of three weeks, I was convinced that E-MYCIN was for real. We could really do it. Great.

We were finished with our experiment. I thanked the pulmonologist for helping us and told him that we're not interested in the rules, in the intellectual property of the rules. That's his knowledge, not ours. He could do whatever he wanted with it. He gave it to the licensing office at Cal Pacific Medical Center, and in 1979, they licensed it to the firm that sold most of these spirometers.

Page forward to 1996. One night I'm in my apartment in Washington, DC. I was the Air Force Chief Scientist. I was having really severe breathing problems. In fact, so bad that I actually called Ted Shortliffe on the phone to ask him, "What do I do? Am I going to die?" He told me what to do, and I survived until the morning. I went over to the Pentagon, and on the fourth floor of the Pentagon, they have a clinic that serves all the generals with stars on their shoulders and the colonels, and me, because I had the equivalent of three stars. I went in there, and sure enough, they had a spirometer in there. I couldn't believe it. The guy sticks me in the spirometer, closes the door, and I breathe in and out. Then I come back out, and he's sitting at his table, typing away. He's looking at the data. I'm sitting there and he has a printer right here, and the printer is printing and printing and printing. I'm looking over the printer, and I can't believe what I'm seeing. I'm seeing my own stuff come out on that printer!

This is 1996! I was shocked! I said to this doctor, “That’s amazing.” He said, “Yeah, it’s amazing. It’s artificial intelligence.” Who knows how many spirometers they’ve sold. I mean, they must have sold thousands of spirometers with that on it.

Friedland: That’s great.

Facts, Domain Experts, and Assumptions

Grad: It is now 12:15. Peter, do you want to prevent us from having lunch.

Hart: Never stand between a hungry group and lunch.

Feigenbaum: Wait, I’m sorry, I missed a piece of the story. I don’t want to use his name, because this has become a public document, but he was so shocked when we said, “Four hundred rules, that’s what you know.” He said, “Is that all I know?” and he took a leave of absence for five years to go do something else.

Lenat: He forgot he also knew those 25 million facts.

Feigenbaum: What was that?

Hart: He forgot that he also knew Doug’s 25 million facts.

Grad: Some portion. My guess is no person knows the 25 million facts.

Lenat: They’re not facts. Assertions.

Grad: Tacit assumptions.

Grad: Living in the world, we know these things.

Haigh: Do experts know much less than regular people then? If we need that many facts to attempt…

Friedland: Not facts.

Feigenbaum: That really didn’t tell the whole story. The original Doug story is that expert systems work because they are in narrow fields. You can accumulate enough knowledge for the system to exhibit excellent behavior within a very narrow range. I mean, there’s a limit in terms

of the number of descriptive attributes or rules that you can get out of a domain. I can give you some examples of things that get out of hand and take a decade to do.

Lenat: Okay, if you narrow the domain enough, you only need 400 rules.

Feigenbaum: You get to the edge. The mantra is "In the knowledge is the power," but once you run out of the knowledge, you run out of the power. As soon as you come to the edge of what the program knows, it can't do anything. For example, we demo'ed MYCIN to John McCarthy. We showed him what was going on and then he typed in, "Did the patient die?" Well, MYCIN knows nothing about death.

It's not in the range, so it just didn't know anything. Doug said, "Well, the thing is that these things are brittle because they don't know enough about the world." You shouldn't fall off a cliff. What you should do is roll down a slope gently. Not fall off a cliff. So, this whole thing should sit on top of 25 million things.

Grad: Let me ask a question. You had these 1,100 specialists sitting there in the background?

Lenat: Yes.

Grad: Are those specialists similar to being in a narrow domain?

Lenat: No, hundreds of them are domain-specific, but most of them are in fact broadly applicable techniques. Some might call them tactics or strategies or stylized inference mechanisms that say, "When you're in this kind of situation, this is a way to make progress."

Grad: Because what I've heard here is, we have the guys building tools and mechanisms that seem to assist people doing the work, but we've also had those that stay in a narrow domain. Obviously, what I did at GE was that. I was trying to design a 39-frame motor! It was a bunch of limits, and it was easy: You could talk to a few people, get the rules down, and it worked. It wasn't 400 rules. But if I were trying to design a motor, even a small motor, I'd be dead. I couldn't even come close to it.

Friedland: Context is all-important. In very narrow contexts of what Ed was describing, which I guess was with obstructive pulmonary diseases, 400 rules seem to incorporate an awful lot of expertise. For a medical specialist, that's true. Whereas, say, a general practitioner is not nearly as deep in any of those things, but probably has a lot more stuff, because that person is looking at the whole body for you.

Grad: I was looking at the book, again, Ed's book about these examples, and there must be, what, 200 examples in there about applications? Somewhere in that ballpark. They're all fenced.

Friedland: Yes!

Grad: That's why you could do it, because you had a mental fence. You were doing just the opposite of what Doug was trying to do and why it's taken him 35 years to get here. We got these things done in months.

Feigenbaum: The generalization of that, that I mentioned this morning, the generalization to all of intelligence is "That is all of intelligence." It's a big switch. If the problem is getting home tonight, you switch to the knowledge-based environment about "getting home tonight."

END OF THE INTERVIEW

