AI: Expert Systems Pioneer Meeting
Session 7: Why Did Expert Systems Decline?

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 Rapaport	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

History of the Computer Science Museum

Burt Grad: I have the great pleasure of introducing again Len Shustek, who is the chairman of the Computer History Museum, now and forever. I think he has a lifetime contract with the museum at zero pay. He created it. He's helped to build it and keeps it going. Len, talk a little bit about the history, would you?

Len Shustek: Sure. I'll tell you how this happened. I've been doing this now for 22 years. I think I mentioned yesterday I got a PhD from Stanford in computer science in the mid-1970s. It wasn't in AI. It was systems, although I did take the AI qualifying exam and passed it because I was terrified of the numerical analysis exam and didn't take that. But I graduated. I went to Carnegie Mellon. I was on the faculty there for a while. I came back to the Bay Area. I did a couple of startups in networking and then went back to teaching.

I was teaching at Stanford. This was 1995, and I taught the same course that I had taken 25 years earlier from Ed McCluskey. Some of you here may have taken that course: EE 282, also called CS w12. It's basically the second graduate course in the design of computers and computer architecture.

Of course, in 25 years some things had changed in the computer business, so the content was different and that was fine. But one of the things that had changed that I didn't find so fine was that history had been erased.

What had changed was that history was erased from the curriculum. When Ed McCluskey taught us about paging, he distributed copies of the papers that had been written by the people who made the applet in the UK. We read the original papers. We learned about the people who had done these amazing things. That wasn't there anymore.

I said, "That's strange." I was a physics undergraduate student. Physics students learn about Newton and Einstein, so why aren't computer science students learning about Babbage and Turing and Von Neumann and all the rest of them.

Now, I wasn't going to change the curriculum. The new textbook had been written by Hennessey and Patterson. Hennessey was on his way to becoming the president of the University, so I had no sway there. I had this crazy idea that my contribution to this problem would be to start a museum. Here we are in the center of Silicon Valley, an important place for computers, why isn't there a museum celebrating the history of computing?

Now, this was a crazy idea because I knew nothing about museums. Absolutely nothing. I started learning by going to other museums. I figured I would visit every computer museum in

the world and discovered there was only one. It was the computer museum in Boston that had been started 15 years earlier by Gordon Bell and Ken Olsen, DEC.

I didn't know anything about it. I went out there and visited. Gwen Bell showed me around. I learned that they were in trouble. They were not a museum that was succeeding for a whole variety of reasons that I won't get into here.

I said that's too bad, but I'm going to go back to Silicon Valley and I'm going to start my museum. I started writing white papers and talking to people about where it would go, and Mike Keller at Stanford tried to convince me that it should go on the Stanford campus.

Then Gordon Bell comes and knocks on my door and visits me over at my office at Network General. I had never met Gordon. Of course, I knew him as a legend. He said, "Do I have a deal for you?" Basically, he was proposing that instead of starting my own museum I would restart what they had originally wanted to do in Boston, which was a history museum, and 15 years later it was morphing into something else. It was becoming kind of a kids' science center. It took me about a millisecond to realize that was exactly the right thing to do.

I joined the board of the Computer Museum in Boston in 1996, with the mission of starting a subsidiary in Silicon Valley called the Computer Museum and History Center. We would restart the original history mission of the museum. And that's what we did.

The first objective was to make sure that all of the amazing artifacts they had collected back there, most of which were not on display, like the Johnniac that somebody talked about yesterday, got preserved. We rented seven 18-wheel moving vans and loaded them up without much legal official permission in Boston and moved them out to California. I managed to secure free warehouse space over at Moffett Field from a friend of mine, and we squirreled that stuff away and started planning and building for our museum.

Housing the Computer Science Museum

Shustek: I started raising money. Started trying to figure out where we would house it. We did an architectural competition for designing a building. NASA was going to give us three acres over at Moffett Field because they were developing a research park there, which they're still developing.

Then the dot-com bust struck. This is 2000, 2001 and suddenly there's a whole bunch of real estate available in Silicon Valley. This was the sales and marketing headquarters of Silicon Graphics. It was an empty building. They had sold it to a developer. We convinced the developer to sell it to us. It was $25 million. Of course, we didn't have $25 million, so I convinced the board to let me borrow $25 million, which we did.

We continued to dream of building a museum and raise money, and we convinced enough people that this was an important thing to do. The overall mission here is something you guys know, that we are living in an extraordinary time. We're living in the step function between the time when there are no computers anywhere in the world and the time when there are more computers than there are people and they're disappearing because they're becoming ubiquitous. We owe it to future generations to preserve the story of this amazing time. That's what this museum is dedicated to doing, and we've had good success.

We bought the building in 2001. We opened the exhibit that I hope many of you have seen downstairs. We have about 40,000 square feet of exhibit space on the history of computing--hardware, software, and all that.

We have considerably expanded the mission of the museum since it started in Boston. In Boston what Gwen and Gordon were doing was basically collecting hardware. Anytime anything that smelled of software would come in Gwen Bell would send it to Gene Sammet of all people. I changed that and started collecting, preserving software in all its forms including source code. We're very keen on doing that.

Computer Science Museum Outreach and Resources

Shustek: We're also expanding the oral histories that we do. We've collected probably 1,000 oral histories. That's the way to get the authentic stories from people who did the work. This is part of that. What Burt has done with these amazing workshops is an incredible contribution to the future historical record for all of this.

We've also expanded into the business arena. We have a center called the Exponential Center, which is focused on entrepreneurship and understanding how companies get built in the computing industry. David Brock heads the Software History Center, which ropes together all of the work we're doing to try to preserve the software history.

We have a big education project. We have kids here. You come here on the weekend, and you will find a whole bunch of middle school kids who are learning about the Eniac in the morning and the Raspberry Pi in the afternoon. We're trying to teach kids, inspire kids to become technologists at a stage in their life, especially girls, where they tend to get turned off from that kind of stuff.

We have a fellow's program where we try to make heroes of the technologists. We just inducted three a couple weeks ago. Ed Feigenbaum was one of our fellows and has been on the board, had him on the board for many years, and he is a contributor.

In terms of operations, we're in the black almost every year. We have about 75 employees, 150 volunteers including people like Ed and Paul McJones and Burt and others. We get about 250,000 visitors to the museum a year, half of them paid visitors to the exhibits and half of them for events, which are either our own events or company events.

We ran out of space in this building. We own this building. It's 7 acres and 120,000 square feet. We're trying to convert as much as possible of this building into public space, exhibits and event space and so forth. All of the other 50,000 artifacts that aren't on display are in a warehouse that we also own over in the East Bay where real estate is a bit cheaper. If you ever want a Raiders of the Lost Ark experience, talk to me and you can get a tour of the warehouse over in the East Bay.

A more recent acquisition is a second building over in East Bay, which is our archive center. That's focused on mostly software but also software documents of all kinds. That's where we do research, where we do scanning, where we try to have visiting scholars come in to use our archive to study.

The machines are great. The physical objects are wonderful for creating exhibits. They're evocative and they make wonderful displays. But the intellectual content of what we're preserving more often is in the paper and the bits than it is in the atoms and the physical objects. We're really committed to preserving the intellectual explanation of this revolution. That's what we're doing.

We encourage your support. The only reason we've been successful and will be successful in the future is because we get support of the people in the computer industry. We're putting as much energy now into understanding what's happening now and what the possibilities are for the future as history. History is a line. It's a bunch of points that you can use to create a line and maybe that line can help you understand what's coming up and what's going to happen in the future.

That's all I have to say. Please support us. Please give us your historical artifacts. They are really important: software, hardware, documents, oral histories, stories, the kinds of things you're doing here today. Support us financially if you can. Become a member. Become a supporter. Put us in your will. It's the computer industry, people in the industry that will continue to make this a success. We are the only people doing this.

Ed LaHay: Was this place's address already 1401 or did you change it?

Shustek: No, we didn't change the address of the building. What Ed is talking about is the IBM 1401. We have a running IBM 1401, in fact, two of them, running IBM 1401 machines

downstairs. We also have a running PDP-1. All of those are maintained and demonstrated every week by our volunteers. The volunteer pools are amazing.

Grad: Ed's question was 1401 Shoreline. Did they get it? Did it start that way, or did they make it happen that way?

Shustek: We did make sure that our phone number was 1010.

CHM Software Preservation

Grad: It's terrific. Again, if any of you are in the area, they do need volunteers. We need people who are interested in software to rebuild our Software Industry Special Interest Group. It's a marriage, we think, between the professional historian kind of work that David and Hansen are doing and those of us who are practitioners. We were the ones who were in the trenches doing the work. David, you missed all the excitement.

David Grier: I know I did. I know I did. I've been renegade. I have a student here.

Grad: We've noted it for the record.

Shustek: Any other questions of me about the institution and what we're doing?

Grad: Software has become a more important element.

Thomas Haigh: You started off with the history not being part of computer science [curriculum]. I'm just wondering if there are any signs that the very impressive accomplishments that you moved onto during the story are having an impact within computer science in terms of making people think that maybe history ought to be something that they're thinking about.

Shustek: I'm not sure we've had a huge amount of success in doing that. We do try, for example, to celebrate failures to make people understand that not everything works, and we hope that may be inspirational. Maybe they can learn from some of those failures. That yes, this brilliant idea they just had is something that's already been tried, and it didn't work. But it's hard to be quantitative about how we've had that effect.

One of the things I have been trying to do is convince venture capitalists to require of all of their portfolio companies to come and take a tour of the Computer History Museum to see the history of the other attempts at what they're doing and learn from that history. I haven't been successful in doing that yet, but maybe I will be.

Grad: What's hard to measure is the impact on individuals and what affect the people who have seen this and come here and how it changes their lives. I keep saying that the Shustek Center, which is what they named the new facility over in Fremont where the software and all the documents are, is really the next generation of the museum.

The first generation you had to attract people, attract money, and the physical exhibits, the hardware and so forth. The newest exhibit they did here, which I hope you'll see for those who haven't, is "Make Software: Change the World!" This was two years ago, I think, just about. They've done a brilliant job. It's just really quite exciting. Its theme is software. That's what makes a difference.

Shustek: There are multiple audiences. "Make Software: Change the World!" is an exhibit about software for the general public. It's a certain level of detail and technology. There's another audience, which are technologists and historians of technology where you can be much more specific and detailed.

Paul and I have been working on various projects to preserve source code. I found the original source code to IBM's APL interpreter, which nobody had, including IBM. I had it in my attic. We've done the System R source code and MacPaint, Microsoft Word, Fortran. PostScript. All sorts of stuff. I'm about to release, for those of you who care, the source code to the Eudora email client. All sorts of stuff.

Grad: Remember, if you guys have software, your initial products, let's preserve it.

Shustek: Right. We want to preserve it. The soft code is the literature of computers. It's the unseen literature of computers.

Model Museums and Exhibits

Doug Lenat: The most impressive museum that I've seen in the last 50 years has been the Newseum in Washington DC. I think there are things from that you may not have thought about. You could actually do analogous things here that would be impactful. By the way, if any of you have not seen it, the next time you go to DC I strongly encourage you to spend a few hours there.

Grad: You could spend a few days there.

Shustek: It's a wonderful museum, but they are mismanaging it. They are spending down their endowment.

Lenat: Yes, I know. Don't copy that part.

Grier: It's anticipated that it will shut down this summer.

Lenat: I know. I encourage everyone to see it while it's still open.

Shustek: Visit it now while it still exists. One of the things I've tried very hard to do here is to run this fiscally conservatively. We have a $30 million endowment. We're in the black every year. We don't build exhibits until we have the money to pay for them. We're running this nonprofit like a for-profit.

Lenat: Yes, don't copy the bad ideas.

Reid Smith: I remember the Nobel Museum. One of the things I remember about it was that they had a kind of an outreach program. They did tours. I live in Houston. They came to Houston for a week. They demonstrated the materials in the museum., Of course, a number of Nobel Prize winners come in to give talks. I'm just curious whether you thought of kind of an outreach away from this building?

Shustek: Well, there are two answers to that. One is everything we do in terms of exhibits and collections is on the web. Our complete catalog is on the web. All of the videos are on the web, thousands and thousands of stuff. Everything in the exhibit is on the web. That's one way we do outreach.

We haven't done physical outreach to other places, satellites, or traveling exhibits. We've talked about a lot. We just got a new CEO Dan'l Lewin who is very interested in doing that, so we are going to put more attention to doing it, but we haven't done it.

Grad: I'm going to break this up. Len, thank you so much for coming and joining us this morning.

Shustek: I'd say this whole experience of listening to you has been fascinating.

Grad: I feel the same way. So magnificently with great planning and forethought and execution on my part, we are now back on schedule.

The Demise of the Expert Systems Companies

Grad: The company discussions ended up taking far more time than I had planned, but seriously these things evolve on their own. They have a life of their own. We have a plan. We have this program. We have the people, we think, to tell the stories. Yet, as it goes along you've got to see what's really happening and what's working.

You guys have been terrific. I've enjoyed it. Now, we can go downhill from here. It's okay. No problem.

We have two other sessions planned for this afternoon. We are going to break actually around 4:00 pm rather than 4:15 pm. For those who wish, the exhibits are open. For those who want to go have a drink, it's before 5:00 pm, so I get a little nervous for you but we'll work it out.

These last two sessions are not well-crafted, yet. You're going to help us craft them as we go along. The first one I have termed the demise of the expert systems industry. But when I listened, maybe the companies demised but not the stuff that you did. Things have changed. Now, I have 25 pages worth of notes here from Ed that I'm supposed to use for the session. Only ten. I'm exaggerating.

Ed Feigenbaum: Well, I gave an AAAI and IAAI joint talk in 1993 called "Tiger in a Cage." The tiger was expert systems, and the cage wasn't going anywhere. There were some reasons why. This is just in the middle of the talk.

Grad: I'm going to ask you guys to come up with each of them. I'll check them against Ed's list. Then any of the ones you don't cover we'll pick up from here. This is a free-for-all, and I would like to encourage some of those who haven't spoken as much to talk.

Grier: Can I add one more thing to your list on this? It's the technology and the business and the others. But the business models: one of the more interesting things that you got started with for ADAPSO was looking at the business models for software firms and all the ones that didn't work. The question is the starting models, and we've heard a couple of them here, business plans for the AI community. Is it more the case that they were the wrong business models?

Grad: That's on the list. Terrific. Tom, go ahead.

Haigh: I want to throw something out there that I hope will be useful for people to react to. I knew nothing about the expert systems industry before this meeting. When I heard there were expert systems companies, I assumed that they would be a bunch of web vans where venture capital got thrown at something or transmitters where there was never a proper business stage. The thing that surprised me, and I mentioned already the parallel with the relational database and management industry coming up through the 1980a, is that most of these companies seem to have had real customers, soft business needs grown to the $10, $20, $30 million of revenue kind of stage. Then things happened to them.

I have listened to the different stories about the things that happened: either they've run into problems and one of the main users takes the technology in-house or it becomes something

that still exists but now it's very domain specific like authenticating financial transactions. It seems like maybe this story that happened is that at the beginning you thought you were engine companies in many cases like a relational database management thing, where you put money into a core technology that's going to be useful in lots of different areas. It seems like mostly you turned essentially into domain-specific services company, where what you were really good at turned out to be understanding this area and making something that works with this thing. That scales in a different way and goes to a different place from really being a very engine-centric cross-platform across the main kind of development path. That's something I've generalized as I've heard the company stories. I'd be interested to see if it's something that you agree with yourselves.

Grad: Go ahead, Hansen, with a further question because you're not going to give an answer, obviously.

Hansen Hsu: I have a related question to Tom's. This is more about something that Peter Hart said. I don't know if Peter is coming back.

Grad: No, Peter will not be.

Hsu: He mentioned something in the break. We were asking him what happened to the knowledge that Syntelligence had acquired in their expert systems. What happened to that knowledge after the company had been sold, etc.?

He realized that there was all this development had happened over the years, but because that was happening in a private company, when the company was sold off that knowledge did not then disseminate back to academia where other companies might draw on it. It was sort of like the knowledge got siloed; the experience got siloed and then sort of disappeared into a vertically integrated company. There weren't new generations of graduate students learning that stuff, so a lot of this knowledge is having to be reinvented. I don't know if other people are having that same story or similar experiences.

Expert Systems Market Size

Grad: Let me start to get the thing going. Paul, you observed the industry over this 10-year period.

Paul Harmon: Well, a couple of points.

Grad: Are these reasons why you feel things didn't continue? Or is this a different story?

Harmon: It relates to why they didn't. One comment is a way to check the market, and the other is a comment on why things don't remain. To check the market, go to AAAI and get attendance figures for the conferences. As a market analyst, the first thing I want to know if I'm talking about a market is how many vendors are advertising at the major conferences. When you start to see attendance and vendors dropping off, you know that the market is closing down for some reason. That certainly happened in that expert systems array.

The second thing is that expert systems knowledge isn't a constant. It keeps changing. In other words, if I'm a human expert I learn things. I go to conferences. I read books. I learn more things. Knowledge bases have to be maintained. If a company had a knowledge base and passed it to somebody else, it would be certainly at the cutting edge. But it would be out of date within a year or two anyway. The ability of these systems to be maintained, to have people bring them up to date is critical, too.

Grad: Once you have that knowledge base system in place don't you naturally keep adding to it, improving it, as it doesn't do anything?

Harmon: That's one of the problems.

Grad: Who owns that? You're supposed to be selling. You're supposed to be getting people to do things. Every time I want you to upgrade the hardware, I come and sell to you. I want you to buy the new version of CICS. I come sell to you. I want you to use the latest version of Microsoft Windows. I obsolete the old one by giving you new features. Did none of you do that? Go ahead, Peter.

Mergers and Acquisitions

Peter Friedland: I think part of the answer to the question is one we haven't talked about yet and is about Silicon Valley companies, in general, or venture backed companies, in general. It's extraordinarily difficult to remain as the same company over a long period of time.

Let's take as a parallel the early biotechnology industry. A similar thing. Explosive early growth because they're doing major technological development. The big four then were Genentech, Biogen, Sidis, and GeneX. One of those company survived. One of those companies found a continuing market and the right niche and the right people to run it as a group.

If you're a venture backed company, which many of these were, you can't stay stagnant. The venture capitalists need to have a buyout. That's either through a successful IPO, which several of the companies did, or through being bought by other companies. After IPO most of these companies, or major pieces of them, were bought by other companies.

So, part of it is the issue of whether expert systems declined in height, which certainly happened. But a large part of it is just the nature of growth companies. You look at any market, and how many have survived more than 10 years with that same name and in that same place? I think a large part of the issue is not unique to expert system companies but to software companies at all.

Grad: I disagree violently.

Friedland: Really?

Grad: The individual companies may have disappeared, but the market has become mammoth. I'd use relational database as one example, you have multi-billion-dollar markets, with company valuations of $40 billion and $100 billion. In desktop publishing, Adobe is the only survivor. But so what? The industry grew. Yours didn't.

That's what sticks in my craw. Why didn't the industry scale? You have a solution to incredibly complex problems, a set of tools nobody else had, Mechanisms are easy to do. People inhouse could do it. Yet, there's no industry.

Feigenbaum: I'll give you one quote. I asked the same question of the CEO of Fujitsu Laboratories in Japan. He's a friend of mine, Sato-san. He was a real fan of all of AI, a real advocate. When he started as an advocate, he wasn't president of the Fujitsu Labs, but by the time I got to him in 1987 he was president. At the end of the interview, I said, “Sato-san, what message would you like me to take back to the people in the US about your experience in expert systems?” He said, "Tell them it's too hard to get the knowledge."

Grad: Okay. Let me challenge that. Here is Schlumberger: inside the company they started doing it themselves. Everybody has knowledge-based systems of some kind that they're using in their products, airlines, trains, everything. Uber. Everybody has knowledge-based systems that they're using, but you didn't get the money out of it. Am I off base entirely? You all seem to be looking blankly at me.

Attracting Business Managers and Executives

Denny Brown: Ed, tell the story about the execs. About attracting software executives.

Grad: Business people.

Feigenbaum: Why is that an answer to his question?

Brown: Why didn’t we make money? Why we didn't successfully make money?

Grad: You didn't have the right business management?

Feigenbaum: Yes. I have a bunch of these slides I gave to Burt. But later on, much later, very recently I was talking all this through with some very savvy business person who gave me another insight into that and said that when you guys got started back in the early 1980s, mid-1980s, AI was not very much known. It was sort of on the fringe. Yes, there were applications you could do, but it wasn't a well-known idea. Therefore, you were not able to attract the first tier of business people to run your company. You attracted second-tier and third-tier business people.

That all of a sudden rang a bell. You mentioned you had to fire somebody. Right, Gary? Somebody else mentioned the same thing. At IntelliCorp we went through three or four CEOs in a few years.

Brown: Five total.

Feigenbaum: Five total. At Teknowledge, it was a rush job.

Lenat: At Teknowledge, we didn't run through enough.

Feigenbaum: No, the first one didn't work out. The second one I was kind of rushed into finding somebody, and I just found the wrong person. He just wrecked the company.

Grad: Brad. Do you have some thoughts? Is this a valid point? That you couldn't attract good business managers.

Brad Allen: Well, I think the thing is that there’s a relationship between the talent you can attract and the lucrative nature of the business. Being able to demonstrate that there's demand gives you the ability to be able to pull in talent of that sort. Right?

After this initial burst of what frankly was government funding, in essence, we were funding on the basis of the hype that was generated by that. It became hard to generate money selling a generic solution.

I think Tom's observation is that when it came time to actually build a business that was sustainable, because we had talked about expert systems as things that would capture knowledge for a very specific domain and deliver value within that domain and helping people make decisions, it created a situation where for people to get success they turned themselves into application companies.

What Peter was talking about with Syntelligence is that kind of an intermediate stage of that. Initially getting into it, the shell was an important thing, but ultimately, they were selling to risk managers. A lot of the financial decision-making stuff, a lot of the things that happen with medical diagnosis, and Ed's example basically became very focused applications.

The companies that survived were the ones that basically pivoted into a particular area. Inference, for example, pivoted into customer relationship management systems and went public and had a reasonable life over that period of time. Other people didn't make that transition because they didn't know how to cross the bridge into something that what kids today would call product market fit.

Grad: How many of you came from academia? Every single one at this table. Do you think that was a factor in why you didn't succeed?

Reid Smith: Tell me what you mean by come from academia?

Grad: You're all PhD's.

Allen: I think it's less about that because there's a lot of people also around this table who had that kind of background and went into networking or database and became tremendously successful in that regard.

Grad: You didn't start as business people. Most all the software companies were started by business people. These were engineering type people. They weren't theoretical.

Allen: I don't know.

Harmon: That's not going to work. I'm sorry.

Grad: I'm throwing things out to get you mad, obviously.

Harmon: First of all, you've got to break the market up into something like large vendors who are selling these expensive tools; mainframe vendors, people who are selling expert systems tools for mainframe; and midsized tools, people like Neuron Data and so forth that are selling tools that aren't on a PC or Macintosh. Just as a first cut. The people who are selling on the PCs and the people that are selling on the mainframes are not academic, but the whole market faded at about the same time. So just being academic is not going to explain it.

There clearly wasn't a market for the large kinds of expert systems that the large companies wanted to build, at least enough of a market that was developing fast enough to keep those companies alive.

Grad: You're saying it's not your fault. You guys did it right, but the market wasn't there.

Expert Systems' Complexity

Harmon: No, I'm not saying that they did it right. I'm saying the market wasn't there.

Grad: Do you agree?

Lenat: I disagree with that. I think that it was closer to what Ed and others were saying: the tools were too hard to use; the education and knowledge transfer to the people who would have to build the systems wasn't there. I was on the Inference advisory board. and I loved ART. It had context and all sorts of wonderful features and truth maintenance. In general, the first thing that the customers would do is turn all that off because it was very complicated. They didn't understand it. They never used it even once. They just used this narrow little tiny iris that wasn't quite enough for them to get enough traction to make it cost-effective.

Friedland: Yes. That occurs all over the place. I look at the Air Force now, and they have a couple of tools that everyone hates using. They're not intelligence amplifiers. They're intelligence sort of subtractors.

The reason that's true is that the Air Force is what Doug said. The Air Force bought braindead versions without the proper initial consulting to mold it into the infrastructure of the company. Therefore, they're being used sort of in a version which the original developers who would get nauseous about if they saw it.

Company Leadership and Market Pivots

Friedland: But I think I somewhat disagree with your disagreement because of what Ed said about people. I mean whether you start with an engineering background or an academic background or an MBA background, it's a rare and wonderful CEO who understands the market well enough and is lucky enough to widen the market trend.

Read the Geoff Moore books, *Crossing the Chasm*, *Into the Tornado*, and so on. Look at what happened to Apple. Apple everyone thought was going to be completely dead. Right? Apple was this tiny niche company that was failing. Stock price down to almost nothing. Then somehow by luck or incredibly good marketing skill Steve Jobs hit upon a trend. Would any of us have guessed 20 years ago that Apple would be the world's largest company in market valuation? And be selling cellphones and equivalent and iPads? And music? No one would've guessed that.

I would claim that independent of how the market changes one thing I've learned and we've learned by firing CEOs and discovering who of us were even semi-competent as a CEO is that it's a rare and wonderful skill. I mean the skill of a Larry Ellison to take Oracle from nothing to being the market leader—really two companies in that market now: Oracle and SAP. That's a rare and wonderful talent. I would contend that if we had been lucky enough to find someone that incredibly good at any of our companies they would still be around.

Grad: The point is that none of you did. None of you found a Larry Ellison.

Friedland: That's right.

Feigenbaum: Gary found Gordon Eubanks.

Gary Hendrix: Yes, Eubanks, I think, did a really good job for us.

Friedland: But he turned it into a completely different company.

Hendrix: He turned it into a completely different company.

Friedland: He saw the market.

Allen: I think that's the thing: the first generation of expert system companies had an extremely skewed image of what the market was because the federal government was coming in and throwing tons of money into the thing. We had relationships with the federal government that we were vectoring on, and it distorted those companies' ideas of what the market was. Literally, we shipped and the next month we were doing a million a month in revenue.

That's a thing that anyone would die for even today. but it was an illusion because it was all early adopters. Once that initial fact-finding effort had gone on within those organizations, they said, "Well, it's either too expensive or we're going to do it in-house."

As Ed said earlier, there was a hubris that had been born of that initial success that those companies had a very difficult time recovering from.

The second thing I think that's important here, too, is it's at a point in time in the delivery of computing technology to people in the market that being able to do that kind of pivot was very expensive: to walk away from commitment to a particular piece of hardware, to go to another target architecture and that kind of a thing.

Today, people just turn it on and off in the cloud, and they're used to timeframes where you just churn them out very, very easily. I think over the last several decades the industry has learned how to make that kind of thing much, much cheaper. And even now, people find it very difficult to do because you have that commitment intellectually to the thing you're trying to achieve. Somebody's telling you, "Well, what we really want is this," and if you can't bridge from where you are to that, then you're dead.

Supply and Demand for Expert Systems

Grad: Look, the software business outside of your area had three elements. Professional services, which seemed to be very profitable, continued to be there a lot for the government. But that's a big chunk. None of you became professional services businesses of any size. A very specialized area but not general.

There are the processing services. I'll do it for you. The cloud is now the perfect example of that. The ADPs and people like made major businesses out of that and continue to this day.

And obviously, the third one is the products, the Oracles and so forth. There aren't many of those left, but they have the products, however they deliver them.

To my knowledge none of you (in AI: Expert Systems) or anyone else has gone into any of those three areas and made an ongoing business out of it of any size. Go ahead.

Fritz Kunze: Well, let me just weigh in here for a moment. I'm going to take Burt's side on this, mainly because it will make for more chaos and confusion. Burt's thesis is that, as I understand it, academic people are unsuited, for some reason, maybe too much hubris or whatever reason, to actually choose a CEO. Instead they say we're victims of not having a CEO. That's what I've heard you all saying.

I want to weigh in on his side here. He may have something there. In my experience, you know, I told you I came from math in an undergraduate background. Everybody was stained with Asperger's. Everybody in my company was an Asperger's victim. I concluded that people who have Asperger's are very good at doing some things that are very detailed, but interacting with humans is not their piece of cake. If I smile at you, Brad, you might smile back at me. But most people who have Asperger's don't have the ability to deal with social cues like that. I would argue—and I don't have a PhD, let me be upfront about that—that people with PhD's typically are way too overdeveloped. They're like a plant that's grown up with very little light. They grew very tall, and unless they are supported by something, they fall over. I'm arguing for a little bit of conflation and confusion here because otherwise it's going to be a very boring, excruciating couple of hours. I think Herb would probably agree with me.

Herb Schorr: No.

Harmon: There's supply and demand. The companies are supplying something. Is there a demand for it? The companies presumably haven't provided it. Is there is still a demand? In other words, are companies out there calling for expert systems now? Are they being developed another way? No.

There are a few cases where they're needed. There are many cases where a simpler type of system is needed. But the idea that every company should have several expert systems running hasn't panned out. It's been decades, and the demand has not bubbled to the surface. The business model was wrong to begin with.

Grad: You're saying there's no opportunity.

Harmon: They might develop the need, but there hasn't been in the last 30 years.

Capturing Expertise

Grier: This is a question that I've been sitting on this whole time. One of the things in studying various computer industries I have found is that very rarely do we see success *ex nihilo* [out of nothing]. Then we come up with something and everyone says, "Oh my God, a spreadsheet. We've always wanted that and we never knew it."

Friedland: You're starting to talk to us in Greek terms.

Grier: I'd like to point out that's technically Latin. Just like I'd like to technically point out Shakespeare never wrote any novels. This is also perhaps the pedigree that makes me an implausible CEO. It's like my most successful book was on how we came up with computers and everyone said, "Oh my God, we have something we can play solitaire on. We've always needed a computer." What did it replace? What work did it replace? It seems to me there's a question here. Why were businesses or organizations thinking it was a good idea to capture expertise and computerize it?

I've been sitting here as we've been going on in looking through the business literature and other discussions about where people start talking about capturing expertise in the 1970s or 1980s. If you look at the business list literature there's virtually nothing about that prior to about 1968 or 1970. Herb has got a couple of things out there about the stages of expertise, which is something I want to get back to, perhaps later in this discussion. But in effect, capturing it, transferring it, moving it across is not there. Drucker talks about the knowledge economy and how knowledge is more important, but he still usually avoids the idea that it's expertise that

you're trying to capture. Peters talks about it in the late 1970s how the World War II generation of leaders is retiring.

How important is it to capture their unique expertise that led industry to an unparalleled success for 25 years, conveniently ignoring a whole bunch of global economic issues behind it? Reagan, in 1980, starts talking about outsourcing government expertise when he becomes president, and you start to see the building up of the Beltway Bandits under him. Yet, there's not a clear consensus anywhere else that I could find this morning or late yesterday about a need to capture expertise.

So, this begs the question for me, what was out there? The narrative we've heard several times is, “This sounded like a cool technology. Let's try it.” The 1990s recession came in and businesses said, “Whoops, can't afford that anymore,” and threw it overboard. Was there a predisposing market for which you were trying to provide a valid service?

Expert Systems versus Knowledge Management

Friedland: There was a reverse of that market in the period from 1995 until the market crash in 2000 called the knowledge management industry. It wasn't expert systems in the sense of expert systems being not just capturing the knowledge but using it for specific applications, but there was a major trend in building what our CTO, Tom Gruber, termed the corporate memory. There were a whole bunch of venture-backed companies—mine was one of them, Introspect Software—that got a lot of money. Folks like the Gartner Group devoted a lot of time to calling that a major industry sector. It was the process of capturing knowledge and context for reuse within large corporations. Companies as big as Citicorp were major buyers into that market.

I'm still not sure whether that market collapsed as a result of the total tech market collapse in 2000 or because corporations decided they would do it themselves without external tools. Did that still survive because people are still buying a lot of things? The inheritors of that industry are things like SharePoint and Livelink and even some of the CRM [customer relationship management] systems like Clarity. That's a very big market still today.

I do know that Introspect Software was about to go public at a $500 million valuation when the market crashed in June. People on our board like Michael Boskin, the Nobel prize-winning economist, said, "Don't worry. We'll wait six months for Morgan Stanley to take you public.” That six months turned out to be forever. I’m not a smart enough economist or marketeer to know exactly what caused that, but certainly what was said here, the recognition of very large organizations that they needed to capture their corporate knowledge somewhere, keeps recurring as a trend.

It was started, I think, as an expert systems idea. It recurred as knowledge management. It's probably recurring today within things, so I don't think the idea was lost as an important idea.

Smith: I was VP of Knowledge Management for Schlumberger. I can tell you that the knowledge management world is alive and well in a number of companies. But, you know, the means of capturing the knowledge and communicating it was the community of practice. It was people. You're right: SharePoint, open text software, even things like Skype were all part of that.

In the end, what happened was tools that were as specialized as Introspect because—remember we talked to Tom and you in the day—that didn't fly. What did fly was human-to-human transfer of knowledge, and that continues.

Grad: Corporate knowledge. Was your market corporations? Was business-to-business your market? It wasn't the consumer?

Brown: For technology, that was correct.

Grad: For everything. The expert systems were things that are being done within businesses to help businesses make decisions.

Friedland: Well, except for things like TurboTax that Ed pointed it.

Grad: Okay. Good example. I was wrong.

Allen: If you go back to the mid-1980s, there was no consumer market. There was no consumer platform for doing computing.

Grad: But there was by the 1990s.

Allen: Yes, that's true. But in the beginning, it was all enterprise.

Brown: Most of this was 1980s. Yes.

Grad: Tom, do you have another question or an answer? We've got too many questions.

Haigh: I'll make an answer then if you prefer,

Burt. It's true, most academics aren't great CEOs. Most tech companies fail.

Friedland: Most CEOs aren't great CEOs.

Haigh: Venture capitalists aren't necessarily the greatest things to deal with, but there are other tech sectors in which venture-funded, academically founded companies have succeeded. Right? Not everyone is a Larry Ellison, but somebody in the relational database management system business was going to be a billionaire, even if it wasn't Ellison. Maybe they got billions, too. I don't even know.

The Sybase SQL Server guys or Informix. Right? The CEO went to prison, as I recall. The opportunity and the market was such that it was going to happen for one of those companies. For some reason, it didn't happen here.

Scalability of Expert Systems

Haigh: Now, something else, I like that thing about Lisp designed by geniuses used by idiots.

Schorr: Sorry, I don't want to be remembered for that.

Haigh: I think you were building software designed by geniuses to be used by geniuses. That goes with this observation that the hard thing is getting the knowledge. If you're Peter Hart or Ed Feigenbaum, you can do amazing things, but there aren't that many Ed Feigenbaum's and Peter Hart's around. What it finishes up being is essentially a services business that's hard to scale because the added value is really coming from being able to do that stuff.

How do you scale that up if you sell the tool? You're saying they turn off all the smart stuff. There isn't a huge base of people out there who are smart enough and have the right background to do the things that the really smart people can.

Schorr: I'm going to echo that a little bit more. One, maybe the technology was premature. I'm throwing out a bunch of hypotheses. My good friend here maybe has proven that. He's got 35 years in this mess. A lot of the tools that made it simple, let's say, took a while. Fortran didn't come overnight either. People were looking at this. They built up from machine coding to assembly language, macros and so on. Backus was a very smart man. Maybe the technology was premature. We needed an awful lot of work. Early people showed great things, but the tools were not necessarily useful.

Okay, the next problem was entrepreneurial talent. I've been associated with a company now, we must be 30 years at it, and we keep puttering along. We make a few million dollars a year. We're very well-liked, but we still have a hard time getting and expanding the base of customers

because basically it's hard to integrate our software with the main software in the company. Nevertheless, we've managed to do that.

But one thing we've never done. Whenever we tried to look for a CEO, no one of any talent is interested because we're too small a business, and it's very hard to sell into these big companies, which is another part of the problem.

Okay. Two more. It's obvious we didn't have a good business model.

Lack of Market Opportunity

Friedland: What company are you talking about?

Schorr: You all. IBM had to do it because the world was doing it, and IBM figured it out. Now, today, they're asleep. At that time, they would pay attention to what was going on in the marketplace, and you had to go into it.

All of this fed the other thing: they were probably undercapitalized. The fact that he was not here, Peter, one year and he sunk because from the sound of it, it sounded like good technology. Where did it go? It should've continued.

You talk about Apple, but that place was a disaster. Somehow, they managed to generate enough money to stay around until Jobs came up with another thing.

Now, I disagree a little bit with the entrepreneurial talent. It's not necessarily business people. The question is whether they found enough entrepreneurs because the great businesses in this country were built by the entrepreneurs. A lot of trouble happened when it went over to the B-school guys in a lot of companies. I won't mention some but nevertheless. Maybe it just didn't happen. There was no entrepreneurial talent who were attracted enough by this to make it, or maybe it's just a bad business model, technology premature, etc. That's what he's telling me: it's taken him 35 years. That's a long haul.

Harmon: There are a couple of occasions where Microsoft, Bill Gates, was asked whether they were interested in AI or expert systems or whether they would be releasing a copy of Lisp. His response was always, "There's not enough of a market for Microsoft to be interested."

Grad: Let's ask that question from the table. He's postulated the fact it was basically lack of market opportunity that really sunk the industry, not any of the other issues that we're raising here. That was the point he made.

Harmon: Just one qualification. I do think that the expert systems introduced computer technology to the industry, in general. I think, it did a great service, but it didn't have a clear customer.

Grad: Not what I'm saying. There wasn't a market for your products or your services or your mechanisms, any of these things because the market just literally wasn't big enough to create a billion-dollar industry. Do you all agree with that?

Friedland: No.

Lenat: No, of course not.

Grad: You don't agree?

Lenat: I don’t think anybody here is going to agree.

Friedland: I don't agree. No.

Grad: Well, then you think Paul is dead wrong.

Lenat: Yes.

Business Management and Growth

Friedland: There's a difference between there being a market and they being smart enough to recognize exactly how to get into that market. As I said, there was even a bigger hyped cycle in the biotechnology companies, and exactly one of them did the right thing. So, I think I would say that none of us or the management teams got it exactly right about how much to spend, what markets to enter at what time, and where to push.

Peter gave the example that they had lots of money, but their CEO spent it like water on all the wrong things. At my last company, our CEO decided because we were riding so high to spend $1 million on signage on our building so that everyone coming down 101 [freeway] from San Francisco could see it and we would attract the best sales guys. That was his approach. I voted against it on the board and was outvoted. I thought it was crazy. Now, who knows if I was right or wrong. In this case, I was right, but I could've been wrong. None of us had the right market teams and things to get it exactly right.

Doug's approach was to basically stay relatively small. Sustain what they were doing. Be funded by the sources that he knew were there and to hang around. Tell me if I'm wrong, but I think you really wanted to hang around long enough to figure out if you were right or wrong.

Lenat: Like the ant and the grasshopper.

Friedland: Yes. Or the tortoise and the hare.

Lenat: Yes. Anyway, it's too glib for us to say we didn't get good enough CEOs or whatever. I think part of it is that instead of trying to sell to industry directly we should have essentially eaten our own dog food. We should have focused on applying expert systems technology to the problem of more effectively building expert systems, building power tools to help extract knowledge and build expert systems like, for example, Randy Davis' Tiresias] system that he did 35 years ago. It was a step in that direction—namely, an expert system to help people build and debug and improve and extend expert systems.

I think if we had more expert-system-based power tools to build expert systems like that then that would have avoided a lot of the problem of us prematurely going out and trying to implant these deployments directly in companies.

Grad: Did any of you start buying up the companies? In almost every other industry, somebody became an accumulator. In the software world, we had Computer Associates. There were three or four companies. like Sterling Software. They bought 80 companies over the course of the years. These companies all grew. Big software services companies, the professional services. A fellow named Bernie Goldstein went out and bought up 50 service bureaus around the country and built a decent-sized business called United Data Centers. Nobody here seemed to do that. Was that a factor?

Expert Systems Tools

Smith: Let me try a hypothesis. Ed has told us many times that in knowledge lies the power, so let me hypothesize this. The hard part or the important part was the knowledge. The tools were a kind of a way of getting there, but they weren't all that great.

Doug's worked on it for 35 years. They're getting better. I saw it from the inside of a company. You say, "Look, what's really important here is the knowledge to solve particular problems." These guys and those guys and these other guys have some tools that may help, but you know the bulk of the work is going to be in getting the knowledge into systems one way or another. Maybe it was the case that the tools that were being offered didn't actually offer that much help toward what the big problem was.

Feigenbaum: Can I finish your speech? For the first part, I gave Burt the title “Knowledge, Knowledge, Knowledge.”

Smith: Yes. I like what Brad said. I'm sorry to interrupt you. I liked what Brad said that it was about solving particular problems, and that's what companies did. Ed knows very well that we wrote a survey of the last 30 years of innovative applications of AI submissions and papers. These are all things that are real in the world, from managing the Hong Kong underground and surface rail system to scheduling in the port of Singapore, to the authorizer’s assistant, which you all remember well. Every time you get called for fraud alerts.

My point is that maybe it was the case that the tools didn't actually offer that much. The real problem, as we’ve all said, was getting the knowledge in there and solving particular problems. What a surprise, in this view of the world, that companies that focused on providing tools in a horizontal way as opposed to a vertical didn't do very well over the long haul.

Allen: The market opportunity was in the applications.

Feigenbaum: Well, the mantra is in the knowledge lies the power; the other side of the coin is it is not the reasoning that is the power. We were selling reasoning, not answers.

Smith: Yes.

Expert Systems versus Business Rules and Procedures

Harmon: Let me give you a side story. In the “Zeroes and Sense,” I covered mostly process. One of the active group of players within the process area are people who do what is called business rules. These are people who specialize; it's a kind of specialty from knowledge management. There are people who specialize in going in and taking the procedures that a company has, rules books and manuals; writing them as rules; and using an inference engine if they're going to use tools, writing them as rules and then enforcing them. These are mostly financial and government organizations that do this. A group like Wells Fargo, last I heard, had a team of 300 people involved in rules across the company. Obviously, it hasn't done them a lot of good.

They had some problems with quality control, but many financial companies and government put lots of money into business rules. Who were the vendors that implemented this? They were all expert system, old expert system vendors.

They had spent the 1990s trying to figure out what to do next. Then somebody who had bought the company or had the company decided business rules was the place to go. They played business rules. There were conferences in the 2008, 2009, 2010. There were big conferences

on business rules. The process vendors, who are mostly flowcharting people, got very excited about expanding the process tools. They mostly bought them, so several of these old expert system tool vendors who then became business rule vendors became part of the business process tool market, and it exists there today.

Grad: How big?

Harmon: There was a demand for rules, and there was a demand for capturing them. They weren't exactly expert systems. They weren't exactly human experts, but they were knowledge.

It's a very easy kind of knowledge to capture. As opposed to interviewing an expert and trying to get them to give you heuristic information, these guys were dealing almost entirely with procedures that had already been stated in a rulebook.

In any case, it's just one variation of what people with this technology went out and found out a niche and did. There was a market for parts of the technology. It just didn't happen to fit the big model.

Technology versus Solutions

Brian McCune: Along those lines, this strikes me as this industry was a technology industry, not a solutions industry. What industry buys, and government to some degree, is solutions. They don't buy technology.

Now, sure, in the first five years, let's try it out, but at some point, I want solutions. The typical thing that I've run into is you talk to a CEO and they say, "What does your product cost? I want to see an ROI [Return on Investment] within one year, sometimes six months. I want all the money I spent on it back, and every year after that, the same amount. If you're going to charge me half-a-million dollars for the prototype and you deliver it, a year later I want to know that I've saved half-a-million and so on." I just don't know that this industry ever got to the level of doing those calculations for its clients.

Expert Systems and the Military

Grier: I'd like to just ask if maybe I've been asking the questions about the wrong vertical. In my work this morning, as I was listening to you all, the one field I found that was deeply concerned about leadership turnover, business rules, knowledge, and the fact that it had a very complex and difficult way of recruiting talent, was the military. There's an early 1980s article by, of all people, Samuel Huntington, the political scientist, about the problems of capturing knowledge and replacing it in high military leadership.

A number of you sat on things and said you couldn't name clients. That's fine, but the question I want to ask is, was there a large military or a large perceived military market for replacing officers? The other part that just dawned on me is that we have, in the military, a very strong large group of people that do nothing but capture military history, military lessons, military rules. Was the military a target market that we have kind of overlooked?

McCune: Well, that was my target market. As we were talking at lunch, there was a discussion at the end of the first phase of this TG computer group in 1989: should we kill off all the AI funding? Are we done with it? They pointed to the logistics applications, which was one tiny part of strategic computing, and they said, "That one application alone saved the Defense Department enough money to pay for all the AI for 25 years." They said, "Okay. Let's keep funding it."

I'll give you another example. I built a system that I can't say very much about. It was a signal analysis system. I built it for $1 million. Five years later the boss of the boss of the boss of my client said, "Thank you, Brian. You saved me $500 million." That's what I'm talking about ROI. He said, "What other systems can you build for me?" I built him two more systems. So, it did pay off for the military.

Friedland: The DoD [Department of Defense] just announced a little while ago the three major strategic technologies for the future of the DoD, and AI is one of them. It's not like the military has ignored it. It hasn't implemented it. I mean Ed was chief scientist of the Air Force for a bunch of years. As he said, one of the reasons they wanted him on board is because they felt this was important.

McCune: Then subsequently Mark Maybury.

Friedland: Yes. It's been hard to implement because of the nature of the military, the nature of federal procurement, and all of those things that surround it.

The last 10 years I've been an advisor to the Air Force Office of Scientific Research and observing the issues of needs and difficulties to implement those things within the workforce. The military is a very large, complex organization with some very fine leaders on the military side but mostly second-rate leaders on what you might call the business side for a very simple reason: business people are attracted by higher salaries. It's a rare person who has the ability to earn a $1 million plus salary in private industry who is going to go and be the CIO, for example, of the Pentagon. Occasionally, there have been very good ones. But most of the time, it's people who don't understand or know how to implement these systems.

As I just said, AI is one of the DoD three key technologies. It's AI, hypersonic, and direct energy, so it's not like the military is ignoring it.

Smith: Just two things. Of course, the Army and the Navy and the Air Force have been very strong in the knowledge management world. The Army center for lessons learned: most of that transfer and codification was done human to human.

Just one other point, do you guys know Peter Hyman? Do you know who Peter Hyman is? Peter Hyman, as you well know, used to work for Schlumberger. All I'm saying is there are guys who are in fairly senior positions these days who also happen to know a good deal about the technology.

Feigenbaum: One follow-up, and I think David knows this part of the answer, but when DARPA chose to respond to the Fifth-Generation Japanese project, they chose two themes, one of which was to promote parallel computing. That didn't work out too well.

The other was to continue the R&D on expert systems by moving it into military domains. That's where they got the big defense contractors involved, Boeing and Lockheed and things like the autonomous land vehicle. If you were a contractor, like we were, you had to work with those people. You had to basically transfer the science that you knew to them.

Expert Systems ROI

Brown: Yes. I was going to respond to Brian's suggestion: did any of us pay attention to ROI calculation in the applications we were building for companies? Burt, it follows up on your point about one of the ways to make money in any industry is professional services. In Teknowledge with the professional services section that we were doing, we had many such successes. One was fraud detection for GMAC. I like to tell the story that we had a 2X ROI 8 months before deployment.

The real story there is that we were doing fraud detection on showroom floor financing for the GM dealerships. The idea was to provide the books and determine whether there's fraud or mismanagement that could lead to a bankruptcy. They had two guys that knew how to do that. They couldn't keep up with the volume, so they hired us, ANS and EDS, to build the application. Just before we got to beta, there was a bankruptcy that cost them $8 million, so they took the beta software, ran it against that, backing up the eight months before running the books and detected the error about a year before.

So yes, we've done some of the ROIs. That was one of the projects that we couldn't talk about. There was a series of that kind of thing where when you take it to an application level you can do ROI. We had another one with Procter & Gamble on a project that I'm not supposed to talk about that had audited 2X ROI in the first 2 years. The plant manager said, "It was easy double that."

We were having success at the application level using the tools that we had in that, but it did take MS [Master of Science] in AI grads from Stanford to implement those and develop the systems. Some of the issues that Doug raised about taking this to a wider audience: it was not easy. Getting the knowledge was difficult.

Expert Systems Support

Grad: Let me ask the question. This is one of the questions that Ed has raised in here. Acquiring knowledge, the knowledge acquisition process is difficult.

Brown: Here's a book. It's *A Practical Guide to Knowledge Acquisition*. This book was written after my run at Teknowledge by two of the people that worked for me at Tek.

Grad: Stay with me. It's difficult, but we have other difficult problems. For example, programming is difficult. Doing complex programming is very difficult, so we found methodologies and broke it down in a certain way that it eventually worked okay. We can build mammoth systems, millions and millions of lines of code, that seem to work. They get buggy, but they're not killer bugs in most cases. Nothing seemed to happen yet. You didn't develop it. Was it just not possible?

Friedland: Well, no, the point you made earlier, I think, is important. If you look at the software companies that have been really successful, the SAPs, the Oracles, and the like, their systems are extremely difficult to implement. That's why if a Citicorp spends $1 billion on buying SAP software, they'll spend $3 or $4 billion with SAP and KPMG and those companies to install it.

Grad: Five times is what we figured for installing and maintaining the system.

Friedland: Whatever the number is. I think one of the fundamental mistakes we made, and everyone has talked about this, is simply having people in for a one-month training course on using these tools. Even if people met and wanted to buy all the tools, I'm sorry as you said. People bought zillions of copies of ART probably about 1985 and right around the tradeshow in Los Angeles. The same thing, with IntelliCorp, people bought Key like chocolate candy bars, but it took really skilled people to use those tools effectively just as it takes really skilled people to use these things.

NASA spent $1 billion on an SAP system and spent almost no money on consulting advice to use it. It took teams of people, and I was on one of the teams, five years to fix the mess that caused at NASA by buying SAP. It ending up with a system that totally ruined its financial management for years. I think if we all had followed that advice early on, we would have done a lot better.

Grad: If anyone of you had followed that advice early on, you could have built a company? This is where you were. You had products. You had the tools. You had the ability to acquire the knowledge. You had the mechanisms for storing and handling it. You had inference engines for using it to run. Every company has a million problems that need solutions, better solutions than they're doing now. Everybody does. Yet, not one of you did it.

The Lack of and Standards in Expert Systems

Brown: Yes. Another point along that line is in contrast with the relational database systems. We didn't have a standards group. We didn't have any industry standards around what the tools were like. What were the representation systems like? Were we all going to be on the same sort of platform for a while? There was very little of that.

Grad: You didn't create a standards group?

Brown: No.

Friedland: Actually, one of us did. I disagree with what you said. One of us did. He's not in the room right now. Monte did it. He built an expert scheduling system. He implemented it successfully at some big companies. Got bought by PeopleSoft for close to $1 billion when they sold Red Pepper. Fortune came out with the 40 under 40 list, and he was the 15th richest person in book value in America right ahead of Michael Jordan in that year.

With the market crash, he was down to only a few hundred million, but Monte somehow did it right. He found a couple of big companies, PeopleSoft, in particular, that just couldn't live without the system.

Allen: But we're not talking about standards in the sense of SQL or something.

Friedland: I'm just saying that somebody did it right.

Grad: No, but standards are good things. We have standards for Cobol. We have standards for Fortran. We had standards for each of these areas.

Brown: We did not, even sort of at the simple level with relational databases SQL as the language.

Grad: Terrible language, but it was the standard.

Brown: It became the standard. It was the de facto standard. We had no de facto standard.

Schorr: Just for history's sake, all of this relational database and SQL had been, what you said, prototyped at the IBM San Jose research lab. Maybe we didn't have enough of that in this area to go to standards. They were not at a standards point. If they had succeeded and been successful, then there would've been standards probably.

Brown: Right. But Oracle, Informix, Sybase, were SQL-based as they were building themselves up. That was, by the way, at the same time; it was the mid-1980s. The mid-1980s was when those companies made their move, but there were standards.

McCune: Those companies were selling products, and those products were replacing existing systems in every company in the world. Expert systems tools were a long way away from doing that.

Brown: We were doing a technology push.

Lenat: There were efforts. I was on the ISO [International Organization for Standardization] standards committee for both rules and ontologies. In general, what happened was either the lowest common denominator because people couldn't agree, or whoever was most persistent, eventually everyone else would say, "I don't want to fight anymore," and then that person would get their way.

Harmon: When was that?

Lenat: That was like about 2009.

Grad: Well, that's now.

Harmon: Not 1987.

Brown: No, no. This is not 1987.

Lenat: No, I'm telling you. That was about 2008, 2009.

McCune: The 1980s were too early to do standards. Standards happen after there's a lot of success, and we need interoperability and compatibility. There was not enough success to start worrying about standards.

Grad: Just a minute. That puzzles me. The minute you had the federal government involved in any of these kinds of things they wanted a standard. They usually made you have a standard. Cobol would not have succeeded without the federal government saying, "You shall use Cobol." They failed on Ada, but with Cobol it wouldn't have happened if it weren't for them.

Allen: Well, as Ed was talking about earlier, Common Lisp was an example of that. But it wasn't material to…

Lenat: That was a Lisp standard.

Brown: Right.

Harmon: There wasn't a need for a standard because there weren't standard activities being done. We're back to that all of a sudden in the mid-1980s a bunch of companies have PhD class software developers who were somehow available to help companies tackle big problems. Like you say, that was in demand.

Was there any need for the tools to build expert systems? Only in a very few cases. There wasn't a market for that. Standardizing the tools wouldn't have helped out. First of all, there were the regular tools, some of them in Lisp, some of them in other languages. There were the mainframe tools. There were the PC tools of various kinds. There were the languages of more than one kind. Europe was entirely on Prolog. It was too early for standards.

AI and Expert Systems Technology in Today's Systems

Feigenbaum: I think one has to go back to the flipside of what Paul said 15 minutes ago. I'll tell you a little story associated with it. The flipside was our work on rules-based systems lives on today, very strongly absorbed into the infrastructure of IT. Somebody mentioned yesterday that if you take off the back cover of some system and you look in there there's a rule-based system in there. The first time this occurred to me it was like 10 years ago. Maybe eight years ago, I did a search on Google, and I got 180,000 hits on the term business rules. It's not the full base. It's not the full inference engine. It's not the full expert system, but the business rules exist.

They also exist in smaller packages of software like the one I saw about a month ago advertised. It was a package to run on your Apple or iPad called Divorce Lawyer. Somebody codified that stuff, and it was selling for $49. You could not get a divorce lawyer by interacting with this thing. If one combed the market, the iOS store or the Android store, you'd probably find 100 of those things called assistants.

Let me tell you this little story, which was prescient. It led to this. I was giving a tutorial; I think it was at one of the AAAI meetings. It was one of these all-day tutorials, and it was in the mid-

1980s. Some Oracle people came to the tutorial. By the coffee break in the afternoon, the Oracle guys said, "Oh, is that all you guys have? Well, we can stick this inside our system. We don't need you guys. We'll just put it right inside our Oracle system." And they probably did, so if you buy an Oracle package you get that.

Kunze: I think the point Peter made earlier about Monte Zweben's companies is a very good one. It's a counterexample to your statement that they're all private companies.

But I did want to make a comment about the standardization product. The government did require that Lisp be standardized as Common Lisp, but it had very little effect on the market, oddly enough. I personally know of a military application of our software built by a company called Sentechnology in Boston. It was used in Desert Storm. I guess in the old days when you tried to figure out the logistics about supplying an army you basically had a big room filled with captains who worked spreadsheets all night long and ate pizza. They figured out how to move a massive amount of material from one location to another. This program was written in Lisp and had the happy ability that when bugs were found you could recompile just little pieces of it and the thing would work just fine. It was written in C, and it would be recompiled on an Amdahl 5890 or whatever the big Amdahl was in those days.

It'd take two weeks before they could test it again. The government couldn't live with that. The government also couldn't live with the fact that they hated Lisp machines and they hated Lisp packs. They demanded it be reprogrammed into C. They spent years trying to do that with great failures. They never got it to work. That's another funny example.

Grier: Just a couple of practical questions on standards. A quick search came up with several that looked interesting: IEEE 1232 exchange of rules and knowledge for expert systems. IEC 2382 rules and knowledge for expert systems became an ISO standard. Oh, blasted. I don't know how to read these numbers. Both are dated 1995.

As standards processes go that would suggest that they got started in the early 1990s. IEC [International Electrotechnical Commission] lists a half-dozen various things, although I don't think many of them deal with expert systems.

Any of these have any impact upon any of you? Were you involved in any of them? Doug?

Lenat: No.

Smith: Has anybody heard of them before?

Lenat: I never heard of it.

McCune: I never heard of it.

Friedland: I just did the experiment Ed suggested. I stopped counting at 400 things labeled as assistants on Android Play. Everything from assistants for navigating the Norwegian airline's webpage to a knowledge base for successful players doing World of Tanks games. Certainly, you're absolutely right within the apps for the cellphones, at least. Probably rule-based assistants live on.

Grad: Isn't that the key point? That what you were doing—the concept of rule-based, knowledge base, whatever terms you want to use—is all over the place. There's literally no company, no product that hasn't this kind of capability built into it itself, but as a separate industry it didn't make it which is fine. I mean it lives on but in a different form.

AI in Videogames

Haigh: Yes. Just leading on from that, I'm a little surprised that nobody has mentioned video games, because video games have a lot of AI in them, all over the place. Is this AI that comes from a different tradition? Is it AI that is craft basis and not academic? I don't know. It seems to me that's where you'd find thousands and thousands of people in the videogame industry just doing AI.

Lenat: No. I guess you're going to talk in the last session about what is called AI today but has almost nothing to do with what we're doing.

Haigh: So, it comes from a different place.

Smith: There are people from the AI community that have spent their time working on videogames.

Lenat: Oh, sure, a few such people like Ken Forbus with Sim City and so on. It's more like there are one or two or three examples. The other thousands have virtually nothing to do with what we're talking about.

Mission-Critical Logic

Grad: I'm going to bring us to a close in a minute. I have here these things that Ed wrote some years ago. Let me just go through quickly to see if any of them trigger something. The first one, I think, we've covered is that knowledge is the base. Knowledge acquisition is the bottleneck, and how you acquire that knowledge seems to be a highly specialized thing requiring the skills of a Feigenbaum or one of these kinds of people. His view is the inference

process itself, acquiring it, is its heuristic and hidden and not a problem for conventional algorithmic processes. It's a narrow niche.

For every one of those things, there are no general solutions. If I read the wording here, if it's too broad it's not powerful enough, and if it's narrow then the user group is too small. So, you're damned if you do and you're damned if you don't kind of thing. The ease of use of these knowledge systems, they have little patience with complexity. These are not easy to use. You don't need a really complex programmer, but it's difficult to make a living out of this. Keep it simple, stupid.

He has two charts on that one. It does matter the cost and speed. These were not easy to use. It wasn't just quick money in most cases, I gather. I'm not reading everything. The knowledge we were saying earlier, the knowledge base you build and spend the money to build is out of date probably the day before you have it operational. It gets worse over time. People are not willing to invest the money to keep it up to date. That's a significant process I gather to do that.

Friedland: That last point I don't think is quite true on the class of applications where you're making sure not to lose precious expertise—you know, we don't want to lose the adhesive supervisor at 3M. I just saw an example at Travis Air Force Base. There's a master sergeant who when he retired, after 40 years, they immediately hired him back as a consultant at much greater cost than his master sergeant salary because if they lost him their scheduling of cargo planes would drop off by 50 percent. Those types of things don't die.

Feigenbaum: Yes. Let me give you a number. In 1987 dollars, to maintain the famous DEC configuration system cost DEC $2.7 million a year because the products kept changing.

Friedland: I believe you. I think in most applications you're right. I was just saying the class of things where there's this guru.

Grad: No, if you have mission-critical logic, people have got to spend the money to keep that logic. Things change. Your missions change. You have to spend the money to maintain it.

Friedland: I agree absolutely.

R&D versus Corporate MIS

Grad: He said no standards were adopted. We certainly talked that through enough. Infrastructure barriers were daunting. Infrastructure was not under the control of the AIKS people. It was a power of the IT people. Some of these people could get in your way. Look, I'm sure we could keep going indefinitely.

Feigenbaum: Let me elaborate on that just a second. Somebody mentioned, Brad, I think, before that all of our companies were living in the world of early adopters. Another way we used to put it was we're living in the world of R&D groups that we're exploring the space. As soon as we ran into the real guys out there who were really going to use this stuff in the real corporation, there was a giant wall. You couldn't penetrate it.

If you tried, you could not sell into an MIS [management information system] group. They would not want to buy anything new. They were so risk-averse. If anything happened to their systems, they were fretting about their jobs. It cost the company $100 million that day if something went down.

I tried to push that wall at Shell using IntelliCorp; we wanted to sell from the R&D group into the real world in Shell. The management information systems guy said, "Are you kidding? It took me so much work to get the CEO to agree to let us put CE on our machine, and you're going to want us to put Lisp on the machine?" They thought I was crazy.

Government Funding for AI and the Japanese Fifth-Generation Project

Colin Garvey: Reading back over a lot of this stuff at kind of a macro level, there seemed to be the promise of expert systems from early stages seemed to be very much tied to the threat that the Japanese were posing in the early 1980s. Brad had mentioned earlier there was a lot of early government money into the field and kind of tied to the threat of the Japanese, a lot of either optimism or competitive sort of spirit there.

I wondered maybe as we're closing this particular session what your thoughts were: as that threat seemed to disappear and the Japanese economy was collapsing in the early 1990s, did that have any effect on the expert systems? Was the threat gone? Did the optimism die down? Did demand die down? Or maybe not at all?

Grad: Money, apparently. Ed, is that something that you want to comment on? As the Fifth-Generation threat disappeared, Japan was no longer scary. Did that mean that the government money dried up and therefore you didn't have the funding?

Feigenbaum: No. Actually, I think it's much more complicated than the picture that you're painting. The people at DARPA who are responsible, who have a responsibility for making significant cases at the DoD level and at the congressional level for getting more money for DARPA, were able to use the Fifth-Generation thing as an excuse for getting some more money. One of the things they wanted to get money for was continuing their very long-term program in AI that Steve Lukasik, the director of DARPA, supported way back in the 1960s when the Arpanet was first being built.

I attended a principal investigator meeting, and Lukasik drew the typical DARPA project. It starts up here in the first year with big funding and then in five years it drops down to zero. Then he drew a horizontal line straight across the blackboard. He said, “That's my AI funding.” It didn't drop down. It just continued on forever. In the 1980s, the Fifth-Generation project was just the excuse for continuing that and getting those applications into military use.

Grier: Ed, can you clarify the rough date of this given that there was great fluctuation in funding in the early 1970s? Was this a meeting in the 1960s? Or was it a meeting in the 1970s? Can you give a date of Lukasik’s comments?

Feigenbaum: It was in the 1960s. It fluctuated a little depending on whether George Heilmeier was in the job or not.

Grier: Right. What I wanted to be sure was that it was not a response to the funding fluctuation of 1970 to 1973.

Feigenbaum: No. No. For example, Tony Tether became director of DARPA. Tony learned about AI from me when I went over to systems control and did that submarine thing. You talked about the over the horizon radar. It was the same thing but for submarines.

Grier: I felt it would be useful to put it into context.

Feigenbaum: Tony said, "Well, let's do personal assistants." It turned out that he allocated over the years $300 million and it went to machine learning. That really funded the revolution in machine learning. DARPA just loves to do that.

Grad: You don't feel that the demise of the Japanese threat led to the demise of the funding of the expert systems?

Feigenbaum: No. Not at all.

END OF THE INTERVIEW

