AAAI-17 Invited Panel on Artificial Intelligence (AI) history: Expert systems

Moderated by:
David C. Brock

Panelists:
Edward Feigenbaum
Bruce Buchanan
Randall Davis
Eric Horvitz

Recorded February 06, 2017
San Francisco, CA

CHM Reference number: X8151.2017

Brock: I'd like to welcome you all to this panel on AI history devoted to the story of expert systems. I'm David C. Brock, an historian of technology and the Director for the Center For Software History at the Computer History Museum in Mountain View, which I encourage you all to visit. Today I'll be acting as your semi-autonomous agent as moderator for the coming discussion. Our panel is also so illustrious that perhaps the briefest of introductions are the most appropriate. Edward Feigenbaum is Kumagai Professor Emeritus at Stanford University and was president of this association (Association for the Advancement of Artificial Intelligence) for 1980 to 1981. Ed was awarded the ACM Turing Prize in 1994 in part for his role in the emergence of expert systems. Bruce Buchanan is University Professor Emeritus at the University of Pittsburgh and was president of this association in 1999 to 2001. Randall Davis is a Professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology and serves on the Executive Cabinet for its Computer Science and Artificial Intelligence Laboratory. He was president of this association for 1995 to 1997. And Eric Horvitz is a Technical Fellow of the Microsoft Corporation where he also serves as the Managing Director of Microsoft Research. He was president of this association in 2007 to 2009.

So to begin, why open a discussion on the history of expert systems now? As the abstract for this session notes, everyone here is quite aware of the current enthusiasm for recognition systems, machine learning, and neural networks in academe, industry, government and the popular imagination. The New York Times recently called today's moment "The Great AI Awakening." Yet this view, and many of us, fail to attend to AI's previous awakenings. In 1985, Allen Newell, this association's first president wrote: "There is no doubt as far as I am concerned that the development of expert systems is the major advance in the field during the last decade…The emergence of expert systems has transformed the enterprise of AI." So today, we'll attempt to open up this history, its transformational aspects and its connections to today's AI awakening by hearing from each of our panelists about their experience of this history and a closing joint discussion. So Ed, could you discuss your work in the first decade of artificial intelligence research and your own intellectual interests that led to the emergence of expert systems in the 1960s?

Feigenbaum: Let me go back to the first generation, 1956-1965. The AI field began with a set of ideas like a Big Bang about the nature of human thought and how to model it by computer. The ideas came from some truly remarkable individuals. Their ideas about problem solving, recognition and learning were fundamental but few. There was a focus on deductive reasoning processes – logic-based or based on heuristic search -- and on the generality of these reasoning processes. In 1962, I wrote about the need to move beyond deductive tasks to the study of inductive processes and tasks which I viewed as dominant in human thought. But it took me two years until 1964 to frame this pursuit in a way that I thought would be fruitful. Being an empirical scientist, I was looking for some people to observe with the idea of modeling their behavior. Now why did I choose to model the thinking of scientists? Because I saw them as being skilled professional induction specialists constructing hypotheses from data and I thought they would be reflectively curious and reductionist enough to enjoy allowing others like me to explore their thought processes.

Brock: Could you tell us, Ed, about some of the scientific experts with whom you worked?

Feigenbaum: In 1964, I was fortunate to find an enthusiastic collaborator, Joshua Lederberg, Professor of Genetics and Nobel Prize winner at Stanford. He too was interested in the question “Can AI model scientific thinking?” So our work together began in 1965 after I joined Stanford. As an aside, Lederberg's mind was one of great vision and insight, one of the top minds of the 20th Century, in my view. But Lederberg was the gift that kept giving. In 1966, Lederberg recruited for us Professor Carl Djerassi, one of the most influential chemists of all time, the father of the Pill [birth control pill] and the head of Stanford's mass spectrometry laboratory.

Brock: Could you talk about in your work with these scientific experts at Stanford, and the development of your first test-bed program for exploring your ideas about induction and expertise?

Feigenbaum: As I said, I'm an empirical scientist, not at theoretical one. I needed a test-bed in which to do these AI experiments. Lederberg suggested the problem that he was working on, inferring hypotheses about organic molecular structures from the data taken by an instrument called a mass spectrometer. Lederberg was doing research for NASA on the design of a Mars probe, designing a mass spectrometer system for detecting life-precursor molecules such as amino acids. In this experimental setting, the test-bed, we could measure, month by month, how well our program -- which was called Heuristic Dendral, or later just Dendral for short -- was performing compared with the performance of Djerassi's PhD students and post docs on the same problem.

Brock: Well, after several years of effort, as we get to 1968, could you describe the kind of fundamental lesson that you had learned from this work?

Feigenbaum: So we proceeded experiment by experiment in this test-bed -- one of the great experimenters of all time is sitting right next to me here [Bruce Buchanan] -- moving toward higher levels of performance in mass spectral analysis, that propelled the movement to higher levels of behavior. What allowed that was the knowledge that we were systematically extracting from the experts in Djerassi's lab. Most of this knowledge was specific to mass spectrometry: including much heuristic knowledge but some was general chemistry knowledge. In 1968, I was writing a paper with Bruce [Buchanan] and Josh Lederberg in which we chose to summarize the evidence from the many Dendral experiments from mid 1965 to mid 1968. It was evident that the improvement in Dendral's performance as a mass spectrum analyst was almost totally a function of the amount and quality of the knowledge that we had obtained from Djerassi's experts, and that it was only weakly related to any improvements that the AI scientists like me and Bruce had made to the reasoning processes used in Dendral's hypothesis formation. So in 1968, I called this observation the “Knowledge is Power Hypothesis.” One data point. Later, as the evidence accumulated from dozens of, or hundreds of, expert systems, I changed the word “hypothesis” to “principle.” The title of the 1968 paper was specifically worded to contrast what we called the Dendral case study with the main paradigm of the first generation of AI that focused on the generality of problem solving. Those of you who are old enough in the audience remember GPS [General Problem Solver]. This was a major paradigm shift for AI research but it took more than five years for the new paradigm to take hold and produce the evidence that Newell was responding to in that quote that David gave you.

Brock: Would you say that…Well, is it your belief that this phrase, "In the knowledge lies the power," holds true today when thinking about AI?

Feigenbaum: Yes, the “Knowledge is Power Principle” is observed in almost all AI applications. For example, in the large number of advisory apps, hundreds that range widely. For example, these are just a few from the last two weeks of the New York Times, the San Francisco Chronicle, and Wired Magazine: divorce law, consumer health advice, planning of travel vacations, income tax advisor and assistant. There was a time that the income tax advisor expert system was the biggest selling expert system of all time. Also, in every one of the justifiably popular AI assistant systems such as Siri and Alexa specifically, people now use the word “skills” to count the specific expert knowledge bases, large or small, that each assistant has. Alexa is said to have many because it is an open system. Siri has far fewer skills. In machine learning R&D, correctly dimensionalizing of the feature space is important, and machine learning engineers use knowledge from experts in making their design choices. That is what we call now “feature engineering.” In some critical applications, for example like car driving, machine learning recognition processes can handle most of the cognitive load but not all. Sometimes, for the so-called “edge cases,” higher-level knowledge of the world will need to be deployed.

Brock: Well, perhaps we could pause for comment or observations from the other panelists at this point. Just to…I would particularly be interested in other people's reactions when they first encountered the knowledge principle or hypothesis, whichever it was, happened to be at the time. Does anybody have a comment?

Buchanan: We'll get to it.

Brock: Okay. Thanks very much, Ed. Perhaps then we'll move on to Bruce. Bruce, could you discuss how the representation of knowledge, and knowledge engineering more broadly, developed through the mid-1970s?

Buchanan: Well, we didn't use the term “knowledge engineering” until the 1970s but we did talk in a 1969 paper that Ed and I were co-authors of with Georgia Sutherland, about knowledge elicitation in AI. It was at a machine intelligence workshop and people there were somewhat stunned that we were talking about organic chemistry. John McCarthy rescued me during a talk by saying to somebody who was giving me a hard time, "Sit down, be quiet, you might learn something."

<laughter>

Buchanan: I forever after loved that man. Well, there were other groups working on knowledge representation at the same time. Harry Pople and Jack Myers at Pitt [University of Pittsburgh] were working with an emphasis on ontologies and mechanisms. Peter Szolovits was working with Dr. Bill Schwartz and that led to a lot of work on the object-oriented frames paradigm. Cas Kulikowski was working on knowledge engineering with Dr. Aaron Safir at Rutgers. There was work in Europe. The LV project came later. There was a lot of isolated work in France replicating some of the early expert systems work, and several projects in France from commercial firms, Schlumberger and Elf Aquitaine being two of the most important. The Japanese Institute for New Generation [Computer] Technology, ICOT, was working on fifth-generation computing largely from a point of view of logic. The French were using Prolog and so did the Japanese. So I think our lesson there, the important part was in coding knowledge. The language you use Prolog or LISP or something else. It didn't matter nearly so much as the paradigm of starting with an expert's knowledge. But we also saw in that time that knowledge engineering could focus on the objects and their relationships in an ontology, a hierarchy. They could focus on the inferential mechanisms that were going on, and in Dendral we were very much interested in what we called the “situation-action rules” at the time. There was an action in the right hand side of the rule, not just a Prolog kind of logical assertion.

Brock: Well as a former student of the subject myself, I was curious to know how your background in the philosophy of science…how did that factor into your contributions to this knowledge engineering area?

Buchanan: I didn't know that about you, I'm glad to know that. Well, I was fascinated with the reasoning process as I'm sure you were. My dissertation was on the process of discovery and trying to orient it into a framework., In the middle of my dissertation, I got to know Ed Feigenbaum in 1963 and began reading the early AI work, mostly by Newell and Simon, and the RAND Corporation publications. And it convinced me that we could make a logic of discovery out of the paradigm of search, a constrained search. So that was the focus within which I got to know Ed and came into this field. So when Ed offered the opportunity to work on Dendral, it was just a godsend because here was an opportunity, one of the early experiments in computational philosophy, to try to do philosophy but with an empirical bent, namely writing programs that would actually produce something that was testable. Then started these discussions with Carl Djerassi's post-doc Alan Duffield and his reasoning process about mass spectrometry and the interpretation of mass spectra was just exactly what I needed in order to instantiate some of those ideas about capturing knowledge, about data interpretation and then, subsequently, theory formation. You've got to, I think, want to contrast this work with other work that was going on at the same time in which people were acting as their own experts. I could not, by any means, claim to be an expert in chemistry or certainly not mass spectrometry. There were other people though like Joel Moses at MIT who was an expert in symbolic mathematics; and Tony Hearn in symbolic algebra; Ken Colby in psychiatry, Todd Wipke in chemistry. These people were also doing knowledge elicitation but it was from their own heads, so it was more like just introspection.

Brock: I was also curious to learn if you could describe what antecedents there were if any about representing the knowledge of a particular domain as rules.

Buchanan: There was a logician who published a paper, Emil Post in 1943, using “production rules” as a complete logical system. That certainly has to be one of the precursors of our work on production systems. Although we weren't following it directly, it was certainly there. Art Samuel's work in the checker player: Art had interviewed experts to understand what is the feature vector and then he did a good deal of reading about checkers, but it was that expertise. And the influential part about that was that his machine learning component-- that once you had the expertise in, in a first order form, it could be improved on automatically. That impressed me a great deal and I always wanted to be able to do that. So we subsequently developed a learning program we called Meta-Dendral that did learn the rules of mass spectrometry from empirical data. A footnote on that. The data were very sparse. It took about one graduate student one year to obtain and interpret one mass spectrum, so we couldn't ask for very much data. This was not a big data problem. And we substituted knowledge for data in that and we continue to believe, I continue to believe, that that's a good tradeoff when you don't have enough data for the big data kind of learning. So just three other things, John McCarthy's paper, "Programs With Common Sense" made a very strong case that whatever knowledge a program was using, it had to be in a form that it could be changed from the outside and that was something Art Samuel was doing with the feature vector weights, but something we were doing with the Dendral rules of mass spectrometry that made a very big difference. Now, Bob Floyd and Allen Newell developed a production rule compiler at CMU [Carnegie Mellon University] and that led to (Ed’s Ph.D. student) Don Waterman's work on representing the knowledge about poker play in a production system. Don's work was extremely influential in giving us the sense that that was the way to do it. And finally Georgia Sutherland had been working with Joshua Lederberg on knowledge elicitation and putting that knowledge into separate tables. They were not rules, they were constraints for the chemical structure generator but they were referenced in a way that they could be changed as data. Those were in my mind the most important precursors.

Brock: At the time and subsequently what limits have you seen in this manner of representation of knowledge as rule?

Buchanan: We saw a lot.

Brock: <laughs>

Buchanan: And our friends at MIT and elsewhere were quick to point out others. We wanted to be testing the limits of a very simple production rule architecture and we knew it was limited, we just didn't know quite where it would break and why. So that was the nature of many of the experiments that we subsequently published in the MYCIN book [‘Rule-Based Expert Systems’ by Bruce Buchanan and Edward Shortliffe] and I would encourage people to take a look. But let me quote from that, "Our experience using EMYCIN to build several expert systems has suggested some negative aspects to using such a simple representation for all of the knowledge. The associations that are encoded in rules are elemental and cannot be further examined except with," some additional text that we put into some extra ad hoc slots. So continuing the quote, "A reasoning program using only homogeneous rules with no internal distinctions among them thus fails to distinguish among several things, chance associations, statistical correlations, heuristics based on experience, cause of associations, definitions, knowledge about structure, taxonomic knowledge," all of those were things that we were failing to capture in the very simple more or less flat organization.

Brock: Well Bruce, could you briefly, if possible describe the fundamental lesson that you took away from this era of knowledge engineering?

Buchanan: Can I do three?

Brock: <laughs> Sure.

Buchanan: There are three different perspectives. From the point of view of computer science, I think the Knowledge is Power Principle is the most important lesson and it's one we certainly have said more than once. At the level of user acceptance, I think the main lesson is that a program needs to be able to explain its reasoning in any decision making situation with high stakes. And third, at the implementation level, the main lesson is flexibility. In the final chapter of the MYCIN book, Chapter 36, it took us a long time to get there…

<laughter>

Buchanan: …we wrote, "If we were to try to summarize in one word why MYCIN works as well as it does, the word would be flexibility. By that, we mean that the designers' choices about programming constructs and knowledge structures can be revised with relative ease and that the users' interactions with the system are not limited to a narrow range in a rigid form." So: knowledge, explanation, flexibility.

Brock: Thank you. Well, Randy, turning to you, could you tell us about what you saw as the most interesting developments in the field that emerged from let's say the mid '70s to the mid '80s?

Davis: I think there were a number of interesting things. One interesting lesson was the value in generalizing the work that had been done. Initially of course, this was the generalization from the individual applications to the so called expert system “shells.” They were put into fairly wide use. Lots of applications got built using them. Not all of these things were acknowledged as expert systems and some of them I think weren't particularly true to the original inspiration and architecture. But the real point is they adopted and spread the ideas-- two good ideas, namely that to be good, a program needed a reasonably large collection of task specific knowledge and second that there were at least semi-principled ways to gather and represent that knowledge. These tools were in some ways analogous to the open sourcing of deep learning tools that are being distributed now and, like those tools, they provide a substantial boost to people who are trying to build these systems. But, as always, it's best if you are one of the anointed ones who know how to use the tools. That's how you get the best use out of them. I think it was true then and I think it's true now.

Another interesting lesson was the way certain insights seemed to echo through the years. We kept seeing the value of explicit readable representations of knowledge using familiar symbols in knowledge representation, expressing knowledge separately from its intended use which of course has already been mentioned. The most immediate consequence of these ideas is to enable multiple uses of the same knowledge, so we had systems that were doing diagnosis with a body of knowledge, explaining the reasoning and the result using that same body of knowledge and then going ahead to teaching somebody with that same body of knowledge, all from a single representation. And just as in when you're building a program, the virtues of encoding something once, saves you from version skew, it was the same thing here in version skew in the knowledge. One of the nice examples of this multiple uses of knowledge came out of the work of Bill Clancy where the basic inspiration was: if we can debrief experts and transfer their knowledge into the program, is it possible to get the program to transfer the same knowledge into the head of a student? That in turn led to lots of interesting work that has already been hinted at in understanding what was insufficient about MYCIN's very simple rule based representation. The systems got considerably more power when that knowledge which was implicit in the rules got explicitly captured and represented in some of the work that Bill Clancy did. Another outcome in that body of work and in other work on intelligent tutoring was the idea that explicit representations of knowledge permits a kind of mind reading, or at least mind inferring. If I have an explicit model of what someone needs to know to accomplish a task and they make a mistake in doing that task, say a diagnosis, I can plausibly ask myself given my model of what they ought to know, what defect in that knowledge would have produced the error that they produced. It's an interesting form of, if not mind reading, at least mind inferring. The final lesson was the ubiquity of knowledge, task specific knowledge. Of course, for example, medicine. Knowledge about debriefing: How do we get the knowledge out of the head of the expert into the program? Knowledge about tutoring: How do we transfer that into the students and knowledge about the general task? Diagnosis as a particular variety of inference. Everywhere we looked there was more to know, more to understand and more to write down in explicit forms.

Brock: Thank you. Could you also now discuss for us your involvement with and your views on the importance of issues of explanation and transparency in artificial intelligence?

Davis: I've been interested in these issues for several decades. The bad news for me at least is after all that time, this session is about two years too late for me to be hailed as visionary because the idea that AI programs ought to be explainable is now in wide circulation. Alas, where were you guys 40 years ago? There's a lot of interest of course in getting understandable AI. There's lots of experiments in getting deep learning systems to become more transparent. As many of you know Dave Gunning has a DARPA program on “explainable AI”, and the overall focus in looking at AI not as automation working alone but as having AI work together with people. All of these things are going to work better with systems that are explainable and transparent. So there's lots of reasons to want this, the most obvious ones are trust and training. Trust is obvious. If we've got autonomous cars or medical diagnosis programs, we want to know we can trust the result. But I think training is another issue. If the system makes a mistake, what ought we to do about it? Should we give it more examples? What kind of examples? Is there something it clearly doesn't know? What doesn't it know? How do we explain it to the system? So transparency helps with making the system smarter. One key issue I think is the representation and inference model. In what sense is the representation and inference model in our programs either similar to or a model of human reasoning? It seems to me that the closer the system’s representations and model of reasoning is to human representations and reasoning the easier it's going to be to bridge that gap and make them understandable.

A kind of counter example of this is currently the vision systems, the deep learning vision systems that are doing a marvelously impressive job of image labeling for example. They're said to derive their own representations and that's great but it's also a problem because they're deriving their own representations. If you want to ask them why they thought a particular picture was George Washington, what could they possibly say? Now the issue is made a little bit worse by the collection of papers these days that show that deep learning vision systems can be thrown off completely by some image perturbations that are virtually invisible to people but cause these systems to get the wrong answer with very high probability. Now the problem is that we don’t know what they're doing and why they're doing it so when you show the system an image that looks to us like a flagpole and it says, "That's a Labrador, I'm sure of it," if we asked them why you thought so, it's not clear what kind of answer they can give us. Now there's been some work in this area of course and to the extent that these systems use representations that are human-derived, they're better off. There's some clever techniques being developed for examining local segments of the decision boundary, but even so, when you start to talk about local segments of a decision boundary in a multidimensional space and hyperplanes, I suspect most people's eyes are going to glaze over. It's not my idea of an intuitive explanation. Now this work is in its very early stages and I certainly hope that we can come up with much better ways to make these extraordinarily powerful and successful systems a whole lot more transparent. But I'm still fundamentally skeptical that views of a complex statistical process are going to do that. Which brings me to a claim that I will make, and then probably get left hung out to dry on, but I will claim that systems ought to have representations that are familiar, simple and hierarchical and inference methods that are intuitive to people. The best test, I think, is simple. Ask a doctor why they came up with a particular diagnosis and listen to the answer and then ask one of our machine learning data systems why they came up with that answer and see about the difference. So let me summarize. If AI's going to be an effective assistant or partner, it's going to have to be able to be trained in focused ways and it's going to have to be able to divulge its expertise in a way that makes sense to the user, not just to the machine learning specialist.

Brock: If I could just ask a follow-on question, in listening to you talk about the importance of both explanation and transparency, it made me wonder, do you think there's a risk that making those requirements of a system might limit them in some way, cut off a potential that they might have?

Davis: I think there's no risk. It will happen and I actually know this from experience. I have a paper in machine learning from last spring that has to do with a medical diagnosis program of sorts where we built the best possible classifier we could in a system that had about a 1,000 dimensional space. It's AUC [area under curve] was above 0.9 and the humans who were doing this task have an AUC of about .75. It was great except it was a black box. So then, working with Cynthia Rudin who was then at MIT, we built machine learning models that were explicitly designed to be more transparent and simpler and we measured that performance and now it's down to about .85. So not only do I know that explanation and transparency will cost you something, we're able to calibrate what it costs you in at least one circumstance. So I think there's no free lunch, but we need both of those things.

Brock: That's fascinating. Any comments or observations from our other panelists at this point?

Horvitz: I'll offer some more general comments in a few moments but, for now, I'd like to underscore the hard challenges and research opportunities ahead on transparency. Working to provide people with insights or explanations about the rationale behind the inferences made by reasoning systems is a really fabulous area for research. I expect to see ongoing discussions and a stream of innovations in this realm. As an example, one approach being explored for making machine-learned models and their inferences more inspectable is a representation developed years ago in the statistics community named generalized additive models. With this approach, models used for inferences are restricted to a sum of terms, where each term is a simple function of one or a few observables. The representation allows people in some ways "to see" and better understand how different observations contribute to a final inference. These models are more scrutable than trying to understand the contributions of thousands of distributed weights and links in top-performing multilayered neural networks or forests of decision trees. There's been a sense that the most accurate models must be less understandable than the simpler models. Recent work with inferences in healthcare show that it's possible to squeeze out most of the accuracy shown by the more complex models with use of the more understandable generalized additive models. But even so, we are far from the types of rich explanations provided by chains of logic developed during the expert systems era. Working with statistical classifiers is quite different than production systems but I think we can still make progress. I'll end my comment there.

Feigenbaum: David, I just wanted to…

Brock: Yes, Ed.

Feigenbaum:, I just want to add on to what Eric just said and I'll change the words a little bit, Eric. But I've been engaged in giving extended tutorials to a group of lawyers at the very, very top of the food chain in law. And the message is: we (lawyers) need a story. That's how we decide things. And we (lawyers) understand about those networks and-- we understand about, at the bottom, you pass up .825 and then it changes into .634 and then it changes into .345. That's not a story. We (lawyers) need a story or we can't assess liability, we can't make judgments. We need that explanation in human terms.

Brock: Yeah.

Buchanan: David, do we have time for one more?

Brock: Absolutely.

Buchanan: A footnote on what Randy mentioned. Randy mentioned Bill Clancy's dissertation and I made Bill just a little bit angry by calling it one of the great experiments with a negative outcome. The hypothesis there was that production rules would indeed be adequate for teaching and that was just plain false and Bill showed not only that it was false but what to do about it then. But I would encourage people to think about the negative conclusions as well as the positive ones.

Davis: And do we have time for one more comment on that?

Brock: Please.

Davis: Going back to what Ed and some other folks were saying about the inadequacies of rule-based systems, absolutely true. And Bruce's list is an excellent one. I don't want to overlook the virtues of trying very hard with simple representation because I think there was virtue to it. It forced us to think very carefully about what the knowledge was, and the taste issue is how long you keep pressing on that because it reveals things and when does it turn into a Procrustean bed. And making that decision point is one of the key things about doing the research properly. Ride it for as long as you can because it helps you see things simply and then throw it away when you realize exactly why it's holding you back. Tßhat's the hard judgment call.

Brock: Thanks. Eric, perhaps we can now turn to you and would you tell us about your work to incorporate Bayesian inference into expert systems?

Horvitz: I'd like to speak as a representative of a small group of folks who were passionate co-conspirators pushing to develop an understanding of the probabilistic foundations of intelligence. Efforts by this group sparked the "probabilistic revolution" in AI over the mid- to late-1980s, a period that some folks refer to as the "AI Winter." Rather than being a time of inactivity, there was a great deal of very active kindling of embers and the sparks created then evolved into a firestorm.

As some background, I came to Stanford University very excited about principles, and architectures of cognition and I was excited about work being done on expert systems of the day. Folks were applying theorem-proving technologies to real-world tasks, helping people in areas like medicine. I was curious about deeper reasoning systems. I remember talking to John McCarthy early on. I was curious about his efforts in commonsense reasoning. In my first meeting with him, I happened to mention inferences in medicine and John very quietly raised his hand and pointed to the left and said, "I think you should go see Bruce Buchanan." And so went to see Bruce and then met Ed [Feigenbaum], Ted Shortliffe, and others. I shared their sense of excitement about moving beyond toy illustrations to build real systems that could augment people's abilities. Ted and team had wrestled with the complexity of the real world, working to deliver healthcare decision support with the primordial, inspiring MYCIN system. Ted had introduced a numerical representation of uncertainty, called “certainty factors,” on top of a logic-based production system used MYCIN. I was collaborating with David Heckerman, a fellow student who had become a close friend around our shared pursuit of principles of intelligence. David and I were big fans of the possibilities of employing probabilities in reasoning systems. We started wondering how certainty factors related to probabilities, measures that we had learned, and that we knew and loved from the past. At one point David showed how certainty factors could be mapped into a probabilistic representation and that helped us to better understand the assumptions of independence being made in MYCIN—and limitations in expressiveness that this would impose. We found that certainty factors and their use in chains of reasoning were actually similar to ideas about belief updating in a theory of scientific confirmation described by philosopher Rudolf Carnap in the early Twentieth Century.

Relaxing the independence assumptions in probabilistic reasoning systems could yield the full power of probability but would also quickly hit a wall of intractability—both in term of assessing probabilities from experts and in doing inferences for diagnosis, based on observations seen in cases. And this led us to start thinking more deeply about methods for backing off of the use of full joint probability distributions and coming up with new models, representations, and languages—some using methods of abstraction and approximation that could provide appropriate expressiveness while reducing the need for so many numbers and so much computation. During this process, I found that mentors and colleagues working on logic-based methods were not particularly enthusiastic about the probabilistic methods. One mentor at Stanford claimed that we had “physics envy” – we were using outdated numerical methods that had been cast aside years before when AI branched away from the fields of cybernetics, operations research, and management science, and those fields were not relevant to making advances at the frontiers of AI. “Those folks really didn't ‘get’ symbolic reasoning and its role in cognition.” Even Herb Simon, who had inspired me deeply, and who I took to be a spiritual guide and mentor, seemed to be skeptical at times. I remember talking with him on the phone and getting very excited about models of bounded rationality founded in probability and decision theory—and a concept I refer to as bounded optimality. “Wasn't this an exciting and interesting approach to bounded rationality?” After a pause, Herb asked me—with what I took to be a bit of disappointment, "So, are you saying you're a Bayesian?"

<laughter>

Horvitz: And I answered, "Yes, I am." My proclamation didn’t diminish our connection over the years but I had the sense that Herb wasn’t excited by my answer to his question. I recall thinking that I’d need more time and would like to explain the ideas better at an in-person conversation. I found rich support outside of AI. I had great conversations with decision-analyst Ron Howard, who had made fundamental contributions in Markov decision processes and decision making, with philosopher Patrick Suppes, who had made contributions to measure theory and probability, and George Danzig, a guru of optimization and founder of the field of operations research. I found they were quite interested in goings on in the world of AI and in the harnessing of probability to solve hard AI challenges. I invited them to serve on my committee, and they joined Ted Shortliffe who represented recent advances in production systems and their application in healthcare. It was an interesting set of advisors and I had a blast working to weave together their different disciplinary perspectives with the goal of building automated systems for decision making under uncertainty—under limited computational resources.

Beyond David Heckerman, I was in touch other graduate students with interest in probability including Greg Cooper, Michael Wellman, Danny Geiger and Ben Grosof, who I see sitting right here at the front of the room, and later Daphne Koller, Moises Goldszmidt, Craig Boutilier, David Parkes, and Nir Freidman. Early on, there was a rising invisible college across several campuses with like-minded graduate students and we eventually connected up with some more senior folks working in AI, including Moshe Ben-Bassat, Peter Cheeseman, and Judea Pearl.

And I have to say this area led to some really interesting work but I want to point out that it was the expert systems tradition, and the aesthetics and goals of that rising field, that really framed the work on probabilistic expert systems or Bayesian systems. For example, we really thought about the acquisition of probabilistic knowledge, how could you do that with tools that would ease the effort, via raising levels of abstraction. The whole tradition of knowledge engineering evolved into methods for acquiring features, relationships, and parameters. The expert systems zeitgeist framed the pursuit as one of working to harness AI to help people to make better decisions. It would have been very surprising to hear in 1985, that we'd be at meetings on AI in 2017 and have folks saying, "We have a new idea; we're going to augment rather than replace human reasoning." In the world of expert systems, this was assumed as an obvious, shared goal--the fact that we would be helping people to work on tasks at hand, whether it be decisions about treating patients or with helping people to understand spectra coming out of a mass spectrometer. And so these notions I think unfortunately have faded with time. We have powerful tools now, but in many ways, folks are only starting to get back to questions about how AI systems should be deployed in ways that help people to solve complex problems in real time. Let me stop there.

Davis: Can I comment briefly on something Eric said?

Brock: Sure.

Davis: I just want it on record, I think we're being taped and videotaped here but I want to acknowledge that he used the word primordial in referring to some of the early work, this is just his disguised way of referring to this side of the table as dinosaurs…

<laughter>

Davis: …and I want that acknowledged. You may continue.

Brock: Thank you. Duly noted. Well, Eric could you continue and talk about how this probabilistic turn changed the development of expert systems per se especially in the areas of biology and medicine?

Horvitz: Well, the first system we worked on with probabilistic reasoning, the Pathfinder system for histopathology diagnosis, getting back to our earlier discussion, had explanation of probabilistic and decision-theoretic reasoning as a distinct focus. This effort was inspired by the work on explanation pursued in studies of expert systems. We really tried to make explanation work. Why did the system make a that recommendation for evidence gathering? What were the expected influences on the probabilities of disease of answering a question about the appearance of tissue through a microscope in different ways? What were the cost and benefits of answering the question. How could we show the value of information by breaking it down into components. So we had explanation running on one of the earliest versions of Pathfinder. We showed how we could shift between text and graphical explanations. We realized that we had a challenge with the fundamental opacity of complex reasoning when the system was computing recommendations for the next best observation. Experts would not get what the system did, because it was doing something unnatural--but more optimal--than familiar human diagnostic strategies. We worked to come up with a simplifying, human-centric abstraction, overlaying a hierarchical ontology of diseases, commonly used by pathologists, onto the reasoning. The modified system was constrained to navigate a tree of categories of disease, moving to more precise disease categories as classes were eliminated. We found that inference was slowed down, with more steps being introduced, but was now more understandable by experts. The pathologists really liked that. So we had this sense that we could overlay humanlike constraints and gain transparency by slowing down the system and potentially introducing a bit more cost in terms of the numbers of observations and tests being asked of pathologists. You'll still get to the right answer.

The work on expert diagnostic systems using probability aligned with the goals of the rule-based expert systems efforts and we started to see the rise of probabilistic systems in the marketplace. But the real change I think in the field happened when it became feasible to store and capture large amounts of data. Back in those first days with the probabilistic systems, we didn't have much data. We had to develop and employ methods that could be used to define and capture conditional probabilities from experts. This was effortful knowledge engineering, similar to the efforts required to capture rules and certain factors from experts. We had to work to assess the structure of Bayesian networks, to lay out the structure of networks and then to ask experts to assess hundreds of numbers, and had to come up with tools for doing that. With more and more data coming available and the rising relevance of machine learning procedures, methods were developed to first mix machine learning and human assessments, and then started to focus more on the data itself in the 1990s. Things have moved away from reasoning deeply about tasks and tracking problem solving as it unfolds and more so to one-shot classification---myopic pattern recognition in a quick cycle, with applications in recommender engines that do one-shot inferences, search engines that use machine learning to do one-shot ranking of list of results, and so on. There’s a huge opportunity ahead, I want to just highlight this, to consider the kinds of problems and the kinds of experiences and decision support that folks were working to provide people with in the expert systems days, but now with modern tools. And I think that that's going to be a very promising area for us to innovate in.

Brock: Thank you. Yes, Bruce.

Buchanan: Eric, you were the one who introduced me to Ron Howard's notion of influence diagrams and that is such a simple and powerful tool for knowledge elicitation, knowledge engineering. You want to say just a word about influence diagrams?

Horvitz: So the whole family of representations that we call graphical models, which include Bayesian networks and influence diagrams, is very expressive. They provide a symbol system for talking about, in a compact way, probability distributions over variables of interest and highlighting critical independencies in the world—independencies among variables that can make our systems tractable and that show promise for capturing what it is we learn when we learn about the world in a way that we can make decisions and inferences. Influence diagrams extend Bayesian networks with representations of actions that can be taken in the world, the outcomes following actions, and utilities that capture preferences over different outcomes. These graphical models offer a coherent way of capturing probabilistic and causal associations and inference machinery has been developed that provides some beautiful capabilities--for example, inference can be called to compute the most valuable information to collect or how to best expend perceptual effort. An area that I pursued in my dissertation work was how to extend similar ideas to guide computation itself. What's the next best thing I should think about as a system? Metalevel decisions about how to allocate base-level computation can be important in systems that exhibit bounded rationality and the methods frame designs and policies for new kinds of cognitive architectures that make use of probabilistic and decision-theoretic reasoning. But the beauty that Bruce was talking about is how these methods enable the graphical, visual expression of their knowledge in an understandable manner, and where people draw arcs, introduce new distinctions and relationships, and have the system propagate uncertainties to infer the probabilities of important variables like diseases present in a patient, based on symptoms, as well as to identify the most valuable information to collect next, and the best actions take in the world. And those kinds of basic functions can be related to tasks that expert systems of the '60s and '70s were aimed at.

Brock: Well to my horror, our panel is-- oh, please, Randy.

Davis: I want to make one more comment, I've always preferred the term, knowledge-based system as opposed to expert system and I like it because it advertises the technical grounds on which the system works: large bodies of knowledge. And I think it's interesting because it holds for people as well as programs. It gets an answer to the question, why are experts, experts, do they think differently than the rest of us, do they think faster than the rest of us? The claim that people and programs can be experts because they know a lot and there's evidence of this in the early work of Chase and Simon who talk about I think it was 30,000 patterns to be a good chess player, more recent work that says, you need to spend 10,000 hours of experience on something to learn to be good at it. There's lots of evidence that knowing a lot is the basis for expertise. And I think that's interesting, it has a not frequently commented on sociological implication. I think it's a profoundly optimistic and inclusive message to the extent that expertise is in fact knowledge based, it becomes accessible to anyone willing to accumulate the relevant knowledge. That's a crucial enabling message in my opinion, perhaps the most important one in education: Yes, you can learn to do this.

Brock: Well thank you, Randy. I was just going to turn to a suggestion that Ed had when we were talking about organizing this panel to close with asking each of you to present to this audience a single sort of takeaway message. I think we may have heard Randy's and I was just…

Davis: Sorry, I'm out of order.

Brock: No. And I just thought perhaps this might be a good time to turn to the takeaway.

Feigenbaum: Okay. I just wanted to make one observation before my takeaway though, which is why I grabbed the mic. David had asked us in advance “To what extent were we motivated by getting behavior that was simply the best in the world, the best there is?” And I would say that we were significantly motivated, although it was unstated much of the time. But we were really after a Dendral that could exceed the capabilities of mass spectrometrists. And in fact, Carl Djerassi did a little experiment with mass spectrometrists around the country to show this. The MYCIN group did an experiment with experts in blood infections around the country, which showed the capability of MYCIN was very good compared to those specialists. I worked on a defense application for DARPA, spent a few years on it, then DARPA gave a contract to MITRE to assess the capability of that system versus the humans who were doing the work in the Defense Department. Our system did significantly better than those humans. As early as 1957, Herb Simon, (you young people may not even know who Herb Simon was, one of the great scientific minds of the 20th Century)

Horvitz: The guy who called me a Bayesian.

Feigenbaum: …made the prediction that a machine would be world chess champion in ten years. Well he was wrong about the time but he was right about an AI program becoming world chess champion. So I think we were significantly motivated, at least I was significantly motivated, by doing programs that did that. To answer your original question, what's my takeaway message? You know, it could be an easy one and I'm going to dismiss the easy one, the easy one of course is that “In the knowledge lies the power." If you take on a job, ask yourself first, where am I going to get the knowledge to do that? This keeps getting referred to on panel after panel, talk after talk, as domain-specific knowledge. But what I really want to say is something that Peter Friedland (one of our former students who worked on a molecular genetics program called MOLGEN) reminded me of: the multidisciplinarity of the work that was going on in the laboratory. In fact, Eric is a beautiful example of that, he's an MD, PhD and we had several MD, PhDs that were among the best students we ever saw, and people interested in a wide variety of fields. Everyone was deeply involved with the details of somebody else's business and that's really exciting--to have a license to do that. So I would say for the young people in the audience, get involved with other disciplines.

Brock: Thank you. Bruce, do you have…

Buchanan: Well that's a hard question but from the point of view of philosophy of science, one of the strong lessons and it was confirmed by one of the great dissertations in AI, namely Randy Davis' dissertation on meta-level reasoning, namely the strategies that scientists and other problem solvers use can be written as knowledge-based systems. The strategy itself is knowledge but it's one level above the domain knowledge. So I take that as one of the very strong lessons to come out of two decades of expert systems work.

Brock: Thank you. Eric?

Horvitz: I would suggest that people today take time to look back at the history, to review the systems that were built, the fanfare of the mid '80s about expert systems and the collapse of that excitement, and the rise of the probabilistic methods that have become central in today's AI efforts. People can learn by understanding the aspirational goals of time and the kinds of systems that were being built in their pursuit. I believe AI researchers will find the architectures of interest, including, for example, the blackboard models--multilayer blackboard models that were developed that employed procedures similar to back propagation, notions of explanation that were considered critical, approaches to metareasoning for controlling inference, and the idea of building systems that engage in a dialog with users, that are embedded with people and situated in a task in the real world and that augment human cognition. These are all key themes of expert systems research, and and some were so fundamental and assumed that we didn't even talk about them, and now they're coming back as new, interesting, and important questions.

Brock: Well, won't you all join me in thanking our panel.

END OF THE INTERVIEW