Learning from Artificial Intelligence's Previous Awakenings: The History of Expert Systems

David C. Brock

■ *Much of the contemporary moment's enthusiasms for and commercial interests in artificial intelligence, specifically machine learning, are prefigured in the experience of the artificial intelligence community concerned with expert systems in the 1970s and 1980s. This essay is based on an invited panel on the history of expert systems at the AAAI-17 conference, featuring Ed Feigenbaum, Bruce Buchanan, Randall Davis, and Eric Horvitz. It argues that artifical intelligence communities today have much to learn from the way that earlier communities grappled with the issues of intelligibility and instrumentality in the study of intelligence.*

If it is indeed true that we cannot fully understand our present without knowledge of our past, there is perhaps no better time than the present to attend to the history of artificial intelligence. Late 2017 saw Sundar Pichai, the CEO of Google, Inc., opine that "AI is one of the most important things that humanity is working on. It's more profound than, I don't know, electricity or fire" (Schleifer 2018). Pichai's notable enthusiasm for, and optimism about, the power of multilayer neural networks coupled to large data stores is widely shared in technical communities and well beyond. Indeed, the general zeal for such artificial intelligence systems of the past decade across the academy, business, government, and the popular imagination was reflected in a recent *New York Times Magazine* article, "The Great AI Awakening" (Lewis-Kraus 2016). Imaginings of our near-future promoted by the World Economic Forum under the banner of a Fourth Industrial Revolution place this "machine learning" at the center of profound changes in economic activity and social life, indeed in the very meaning of what it means to be human (Schwab 2016).

Far too often, these pronouncements and perspectives fail to attend to artificial intelligence's previous awakenings. Over 30 years ago, in 1985, Allen Newell — one of the key figures in the emergence of artificial intelligence as a field in the 1950s and the first president of the Association for the Advancement of Artificial Intelligence (AAAI) — 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" (Bobrow and Hayes 1985). This article frames and presents the discussion at an invited panel, "AI History: Expert Systems," held at the AAAI-17 conference in San Francisco, February 6. The panel's purpose was to open up this history of expert systems, its transformational aspects, and its connections to today's "AI awakening" (Brock 2017).[1]

The history panel featured four key figures in the story of expert systems and was moderated by the director of the Center for Software History at the Computer History Museum, David C. Brock. Edward Feigenbaum is the Kumagai Professor Emeritus at Stanford University. He was president of AAAI in 1980–81, and he was awarded the ACM Turing Award for 1994 in part for his role in the emergence of expert systems. Bruce Buchanan is a university professor emeritus at the University of Pittsburgh, and he was president of AAAI in 1999–2001. Randall Davis is a professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology. He was president of AAAI in 1995–97. Eric Horvitz, MD, PhD, is a technical fellow of the Microsoft Corporation, where he also serves as the managing director of Microsoft Research. He was president of AAAI in 2007–09.

Perspectives from two historians of science and technology provide a very useful framework for approaching the history of expert systems, and the discussion at the 2017 history panel. Michael Mahoney, a history professor at Princeton University, was a particularly influential figure in the study of the history of computing. In a 2005 article, "The Histories of Computing(s)," Mahoney presented a concise statement of several of his most fundamental insights from his many years of study in the field. "[T]he history of computing," he wrote, "is the history of what people wanted computers to do and how people designed computers to do it. It may not be one history, or at least it may not be useful to treat is as one. Different groups of people saw different possibilities in computing, and they had different experiences as they sought to realize these possibilities. One may speak of them as 'communities of computing,' or perhaps as communities of practitioners that took up the computer, adapting to it while they adapted it to their purposes" (Mahoney 2005).

For Mahoney, a defining activity of these various communities of computing was creating software that, for him, constituted the modeling of certain features of the physical or social world. Making software for Mahoney was putting the world into the computer: "[Software] Design is not primarily about computing as commonly understood, that is, about computers and programming," he explained, "It is about modeling the world in the computer … about translating a portion of the world into terms a computer can 'understand' … [P]utting a portion of the world into the computer means designing an operative representation of it that captures what we take to be its essential features. That had proved, as I say, no easy task … If we want critical understandings of how various communities of computing have put their portion of the world into software, we must uncover the operative representations they have designed and constructed" (Mahoney 2005).

An historical expert on a very different subject — the Scientific Revolution of the 16th and 17th centuries — and a history professor at Cornell University, Peter Dear provides an account of the two fundamental purposes toward which scientific and technical communities, including Mahoney's communities of computing, direct their activities: *intelligibility* and *instrumentality*. Crudely summarized, Dear proposes that there are two distinct, separate, but intertwined purposes that have motivated these communities. One is a pursuit of the "intellectual understanding of the natural world," including ourselves. This is the striving to make sense of the world, to provide satisfying answers to basic questions about how things are, and why they are. Dear notes, "Evidently … there are not timeless, ahistorical criteria for determining what will count as satisfactory to the understanding. Assertions of intelligibility can be understood only in the particular cultural settings that produce them." The other purpose is the creation of effective techniques that afford, as Dear puts it, "… power over matter, and indirectly, power over people." Here the goal is the creation and refinement of an "operational, or instrumental, set of techniques used to do things … Such accomplishments … in fact result from complex endeavors involving a huge array of mutually dependent theoretical and empirical techniques and competences" (Dear 2006, pp. 1–14).

Both goals of intelligibility and instrumentality can clearly be seen in the community of computing — perhaps, more properly, communities — involved in artificial intelligence. On the side of intelligibility lie questions about our understanding of human intelligence: how is it that we reason, learn, judge, perceive, and conduct other mental actions? On the side of instrumentality reside myriad activities to create computer systems that match or exceed human performance in tasks associated with the broad concept of "intelligence." This instrumental dimension in the history of artificial intelligence is of a piece with a major through-line in the history of comput-

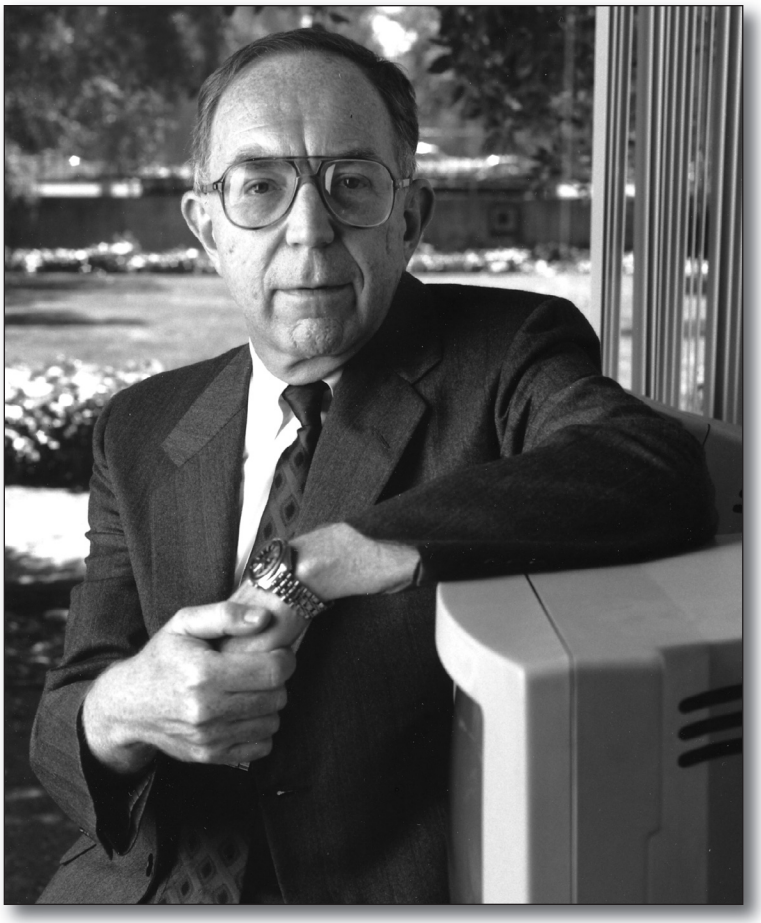
ing more generally, in which scientists and engineers developed machines to, at first, aid human practices of mathematical calculation, but these machines quickly came to exceed human intelligence's unaided capacity for calculation by many orders of magnitude. From one angle, the pursuit of instrumentality in artificial intelligence may be seen as an effort to extend this surpassing of the human capacity for mathematical calculation to additional capabilities and performances.

It is nonetheless very clear that intelligibility was an enormously motivating goal for the emergence of artificial intelligence. In his reflections on the history of artificial intelligence to 1985 cited above, Allen Newell also powerfully surfaced the importance of intelligibility to the artificial intelligence community of which he was a part: "One of the world's deepest mysteries — the nature of mind — is at the center of AI. It is our holy grail. Its discovery (which will no more be a single act than the discovery of the nature of life or the origins of the universe) will be a major chapter in the scientific advance of mankind. When it happens (that is, as substantial progress becomes evident), there will be plenty of recognition that we have finally penetrated this mystery and in what it consists. There will be a coherent account of the nature of intelligence, knowledge, intention, desire, etc., and how it is possible for the phenomena that cluster under these names to occur in our physical universe." (Bobrow and Hayes 1985, 378)

As Ed Feigenbaum explained in the AAAI-17 panel on the history of expert systems, and in terms that directly echo Mahoney's view of software as modeling, the roots of expert systems begin in the first decade of the AI community's pursuit of intelligibility:

> 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* [emphasis added] 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* [emphasis added] 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.

Allen Newell and Herbert Simon created perhaps the first artificial intelligence program, Logic Theorist, in 1955–56 as a model of deductive reasoning, and also as a kind of self-modeling of their own familiarity with creating proofs in formal logic and mathematics. That this model reproduced a set of proofs created by Bertrand Russell and Alfred North Whitehead in their seminal *Principia Mathematica*, and even arguably improved upon one, was taken as strong confirmation of their model.



AAAI Archive File Photo.

Edward Feigenbaum.

In contrast, Feigenbaum's interest was in modeling the inductive reasoning that he believed was vital in human intelligence generally. From his study of and with Newell and Simon, Feigenbaum drew the lesson that such modeling of human reasoning in the computer needed specificity. He believed that a particular "test bed" was required to exercise and refine the model, and to draw conclusions from it. After extended deliberation, Feigenbaum decided that "professional inducers," people who were paid to make inductions, would make for a productive test bed. He would model, therefore, the reasoning of empirical scientists. To his surprise, Feigenbaum found that a prominent empirical scientist, indeed one of the world's leading geneticists, shared his interest in the possibilities for computational models of scientific reasoning:

AAAI Archive File Photo.

Allen Newell and Herbert Simon.

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.

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 postdocs on the same problem.

Throughout the 1960s, Feigenbaum — in collaboration with Bruce Buchanan and others — continued to evolve the model of the organic chemists in Carl Djerassi's laboratory, and in particular their capability to interpret the data about particular sorts of organic compounds from their mass spectrometry instrumentation. This modeling had two basic features. For one, the artificial intelligence researchers developed the model of inductive reasoning processes in DENDRAL. In addition, they modeled the organic chemists' knowledge as a store of rules, roughly in the form of "If, Then" statements.

Photograph Courtesy, National Library of Medicine

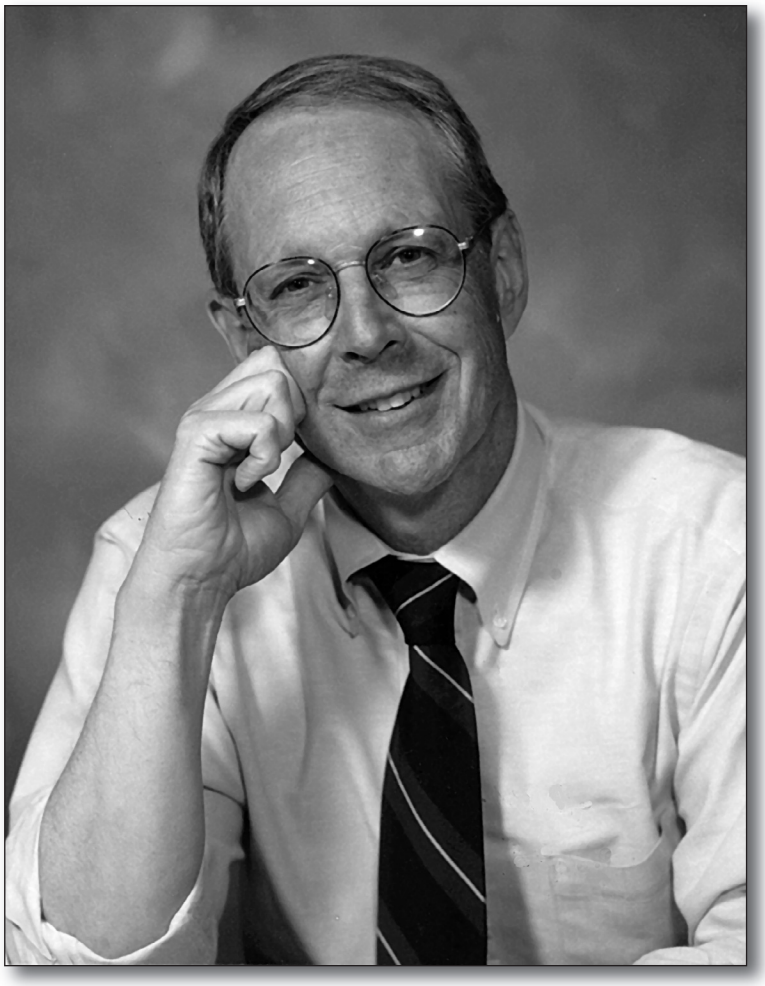
The Original Dendral Team, Twenty-Five Years Later.

The development of DENDRAL along these two lines served both intelligibility and instrumentality. On intelligibility, the motivating question was how to explain different abilities in human reasoning. How is it that experts render better judgments? Is it that some people possess a fundamentally superior way of reasoning than others? Or is it rather that accumulated, organized experience is the key, with people reasoning in more or less the same fashion?

Research in the nascent field of expert systems sought to address the question of special reasoning versus accumulated knowledge as the basis for expert judgments. With this also lay the promise of instrumentality. If the key to expert performance could be unlocked by investigations of a computer system, might such a system come to match and then exceed the performance of any human expert, as was already the case with mathematical calculation? This co-implication of intelligibility and instrumentality was not lost on the primary sponsor of artificial intelligence research in the United States, at least until very recently: The US Department of Defense's famous Defense Advanced Research Projects Agency (DARPA).

As Feigenbaum recalled, by 1968 he and his colleagues were prepared to draw conclusions from work on DENDRAL, from what the performance of their model of this human expertise had shown them. The conclusions were, themselves, an induction from a model of chemists' expertise in the interpretation of mass spectra of particular families of organic molecules:

> So we proceeded experiment by experiment in this test bed … 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



AAAI Archive File Photo.

Bruce Buchanan.

> 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.

Continued modeling of human experts, and in particular scientists and engineers, led to expert systems that, for very specific kinds of expertise, could meet and even exceed the performance of human experts. This achievement of instrumentality — a novel capability to do things, namely to exceed some performances of human experts — eventually led to a great enthusiasm for expert systems within the artificial intelligence community and its military patrons, then quickly drawing in corporations, investors, entrepreneurs, and the popular press.

Yet the route to these enthusiasms was painstaking work along the same two developmental lines for modeling human expertise as computer systems: changes to the inductive reasoning processes and also to the representation of expert knowledge and the means of making that representation. Bruce Buchanan concentrated his efforts on the latter, which, he explained eventually became known as *knowledge engineering:*

> 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 coauthors 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." 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 [the 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 … 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.

For Buchanan, as with Feigenbaum, the motivation of intelligibility was, at least initially, primary for the development of expert systems. Buchanan recalled:

> Well, I was fascinated with the reasoning process.... My dissertation [on the philosophy of science] 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* [emphasis added] — 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 postdoc 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 [and his colleagues] 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.

As Buchanan showed, modeling the expertise of others as opposed to introspective self-modeling did not fully distinguish the subfield of expert systems from other areas of artificial intelligence work. Rather, the development of expert systems relied on mixtures of both kinds of modeling.

Whether from self-modeling or modeling of others, Buchanan and others created a particular kind of representation of the modeled knowledge known as *production rules,* a system of "If, Then" statements. Buchanan explained:

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 on the checker player: Art had interviewed experts to understand … the feature vector and then he did a good deal of reading about checkers.… And the influential part about that was … his machine learning component — that once you had the expertise in, in a first-order form, it could be improved … 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 continued 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 … that was something Art Samuel was doing with the feature vector weights, but something also 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 [Feigenbaum's PhD 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.

This is not to say that Buchanan and others believed that these production rules were the last word in modeling human expert knowledge in a computer. When asked if he believed that the representation of knowledge as a rule had limitations, Buchanan replied, "We saw a lot." He continued:

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.

The modeling of human experts' knowledge in expert systems as production rules was provisional, intended to reveal what kinds of performance they could produce and what they could not.

From Buchanan's involvement with knowledge engineering into the middle 1980s, he drew three fundamental lessons:

AAAI Archive File Photo.

Randall Davis at AAAI-92.

> 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 … 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.

These three issues — knowledge, explanation, and flexibility — have also become central to contemporary discussions of multilayer neural networks and machine learning, with "knowledge" now taking the guise of the datasets used for training, and "flexibility" now largely couched in terms of the fragility or brittleness of machine learning systems. Explanation, or the lack thereof, however remains a key challenge for today's artificial intelligence efforts.

Randall Davis, who has placed explanation at the center of his work in artificial intelligence, saw the development of expert systems from the middle 1970s to the middle 1980s continue to evolve the two main strands of development that had been present since the start: the reasoning processes and the representation of expert knowledge. Much of that continued development, in Davis' view, was in the direction of generalization:

> 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.… 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 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 … 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.

These matters of rendering implicit knowledge explicit, of mind inferring, and of knowledge transfers are all of a kind with Davis' concern for explanation and transparency in artificial intelligence. He explained:

I've been interested in these issues for several decades. The bad news, for me at least, is after all that time … 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 are to human representations and reasoning, the easier it's going to be to bridge that gap and make them understandable.

A kind of counterexample 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.

For Davis, greater fidelity in the modeling of human expertise into AI systems should serve both intelligibility and instrumentality.

And yet, as Davis underscored, intelligibility — explainable AI — also comes with some instrumental cost. Asked if the requirement for explanation and transparency could limit other aspects of performance in an AI system, Davis answered:

It will happen, and I actually know this from experience. I have a paper in *Machine Learning* from last spring [March 2016] 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. Its AUC [area under curve] was above 0.9 and the humans who were doing this task have an AUC of about 0.75. 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 0.85. 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.

AAAI Archive File Photo.

Eric Horvitz.

Eric Horvitz, a key figure in the statistical and probabilistic turn in artificial intelligence research, shared this same vision of the importance of explanation especially in contemporary work:

> 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.

Feigenbaum too stressed the importance of explanation — intelligibility — not just in the motivations behind artificial intelligence systems, but also, with Davis and Horvitz, as part of their instrumentality, their value in use:

> 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.

While Horvitz is most associated with the statistical turn in artificial intelligence that is seen as adding profound new challenges to explanation and transparency, his route to this stance was through his engagement with and deep interest in expert systems. Horvitz explained:

> I came to Stanford University very excited about the 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 [I] 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 in 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 … David showed how certainty factors could be mapped into a probabilistic representation … 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 20th century.
>
> Relaxing the independence assumptions in proba-

bilistic reasoning systems could yield the full power of probability but would also quickly hit a wall of intractability—both in terms 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.…

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?" 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.…

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.

Despite these continuities with the interests, ethos, and some of the central issues of the tradition of expert systems, the "probabilistic revolution," as Horvitz calls it, had real consequences for the subsequent development of expert, and other, artificial intelligence systems. Horvitz recalled:

The first system we worked on with probabilistic reasoning, the Pathfinder system for histopathology diagnosis … 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.…

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.…

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.

In reflecting on the importance of the history of expert systems for the communities of artificial intelligence today, Ed Feigenbaum stressed the importance of instrumentality as a motivation:

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 (… young people may not even know who Herb Simon was, one of the great scientific minds of the 20th century) made the prediction that a machine would be world chess champion in 10 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.
>
> [T]he "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 … 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.

For Bruce Buchanan, the primary lesson from the history of expert systems is that the very reasoning strategies, the thought processes, used by human experts are themselves forms of knowledge that can be learned, acquired:

> 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 metalevel 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.

Randall Davis shared this very same perspective, even going so far as to suggest that "knowledge-based systems" would have been a preferable term to "expert systems." He explained:

> 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 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.

For Eric Horvitz, his entreaty for the contemporary artificial intelligence community is for it to look at the history of expert systems, their technical character, and the conclusions they supported as a resource for addressing today's concerns. He concluded:

> 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 backpropagation, notions of explanation that were considered critical, approaches to metareasoning for controlling inference, and the idea of building systems that engage in a dialogue 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 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.

To date, a pronounced pattern in the history of artificial intelligence is that of oscillation. The communities of artificial intelligence have swung their attention to and from a core set of interests and approaches repeatedly: heuristic problem-solving, neural networks, logical reasoning, and perception. Each has fallen into and out of, then back into, favor for at least one cycle, some more. Yet many within the artificial intelligence community see steady advance. As one recent report put it: "While the rate of progress in AI has been patchy and unpredictable, there have been significant advances since the field's inception 60 years ago. Once a mostly academic area of study, 21st-century AI enables a constellation of mainstream technologies that are having a substantial impact on everyday lives." Even so, outside the

artificial intelligence community, the broader academic, commercial, governmental, and cultural interest in artificial intelligence has oscillated from almost-exhilaration to near-despair several times.

It would seem that this pattern of oscillation is, to some degree, due to the very subject of artificial intelligence: the broad and, in many places, nebulous concept of intelligence. Intelligence encompasses the "fast thinking" of perception to the "slow thinking" of complex problem-solving. It ranges from "deep learning" to "deep thinking," and combinations thereof. Given such range, it is unsurprising that a field would shift its attention from one area to another, as certain lines of inquiry gain traction and others appear stuck. But the pattern of oscillation and the sweep of intelligence pose the question: Whither integration?

Can models of problem-solving be integrated with models of perception? Can models of recognition be integrated with models of reasoning? What is the role of knowledge, especially in the guise of common sense? Is a more integrated model of human intelligence necessary for both greater intelligibility and greater instrumentality in artificial intelligence?

Notes

1. A video recording of the panel has been archived by the Computer History Museum (Mountain View, CA) under the title "AAAI-17 Invited Panel on Artificial Intelligence history: Expert systems." Catalog Number 102738231. www.computerhistory.org/collections/catalog/102738236

.

References

Bobrow, D. G., and Hayes, P. J. 1985. Artificial Intelligence — Where Are We? *Artificial Intelligence* 25(3): 375–415.

Brock, D. C., moderator. 2017. AI History: Expert Systems. A Panel held at the 31st AAAI Conference on Artificial Intelligence. Panelists: Edward Feigenbaum, Bruce Buchanan, Randall Davis, Eric Horvitz. San Francisco, February 6. Palo Alto, CA: Association for the Advancement of Artificial Intelligence. videolectures.net/aaai2017_sanfrancisco/.

Dear, P. 2006. *The Intelligibility of Nature: How Science Makes Sense of the World*. Chicago: University of Chicago Press.

Lewis-Kraus, G. 2016. The Great A.I. Awakening. *The New York Times Sunday Magazine,* December 14. www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html.

Mahoney, M. S. 2005. The Histories of Computing(s). *Interdisciplinary Science Reviews* 30(2): 119–35.

Schleifer, T. 2018. Google CEO Sundar Pichai Says AI Is More Profound Than Electricity and Fire. *Recode,* January 19. www.recode.net/2018/1/19/16911180/sundar-pichai-google-fire-electricity-ai.

Schwab, K. 2016. The Fourth Industrial Revolution: What It Means and How to Respond. *World Economic Forum,* January 14. www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/.

Suggestions for Further Reading

Brock, D. C., moderator. 2017. AI History: Expert Systems. A Panel held at the 31st AAAI Conference on Artificial Intelligence. Panelists: Edward Feigenbaum, Bruce Buchanan, Randall Davis, Eric Horvitz. San Francisco Hilton, February 6. Palo Alto, CA: Association for the Advancement of Artificial Intelligence. archive.computerhistory.org/resources/access/text/2018/03/102738236-05-01-acc.pdf.

Buchanan, B., and Shortliffe, E. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison Wesley. people.dbmi.columbia.edu/~ehs7001/Buchanan-Shortliffe-1984/MYCIN%20Book.htm.

Buchanan, B.; Sutherland, G.; and Feigenbaum, E. A. 1969. Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry. In *Machine Intelligence Four: Proceedings of the Fourth Annual Machine Intelligence Workshop,* edited by B. Meltzer and D. Michie. Edinburgh: Edinburgh University Press. profiles.nlm.nih.gov/ps/access/BBABKI.pdf.

Davis, R., and Lenat, D. 1982. *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill Advanced Computer Science Series. New York: McGraw-Hill.

Feigenbaum, E. A.; McCorduck, P.; and Nii, H. P. 1988. *The Rise of the Expert Company*. New York: Crown. stacks.stanford.edu/file/druid:qf857qc1720/qf857qc1720.pdf.

Heckerman, D. E.; Horvitz, E. J.; and Nathwani, B. N. 1992. Toward Normative Expert Systems: Part 1. The Pathfinder Project. *Methods of Information in Medicine* 31(2): 90–105. www.microsoft.com/en-us/research/wp-content/uploads/2016/11/Toward-Normative-Expert-Systems-Part-I.pdf.

McCorduck, P. 1979. *Machines Who Think*. San Francisco: W. H. Freeman and Co. archive.org/details/machineswhothink00mcco.

Nilsson, N. J. 2010. *The Quest for Artificial Intelligence*. New York: Cambridge University Press. ai.stanford.edu/~nilsson/QAI/qai.pdf.

David C. Brock is the director for the Center for Software History at the Computer History Museum in Mountain View, Calfornia. A historian of technology, he recently coauthored *Moore's Law: The Life of Gordon Moore, Silicon Valley's Quiet Revolutionary* (Basic Books, 2015). At the Center for Software History, Brock is leading efforts to preserve the history of artificial intelligence. If you have materials for possible donation, or share this interest in history, please email dbrock@computerhistory.org