18

Consulting Systems

18.1 The SRI Computer-Based Consultant

As my colleagues and I at SRI cast about for ways to continue our planning and vision research we had been doing under the "Shakey the Robot" project, while satisfying DARPA's interest in militarily relevant applications, we hit upon the problem of equipment maintenance, repair, and training. We pointed out that any technology that could reduce expenditures for these items and lessen the need for utilizing scarce human experts would be extremely important to the military. Furthermore, we said, this need "cannot be satisfied merely by writing more and better manuals. A sophisticated computer system seems to us essential."

We proposed to begin work on this problem by developing an environment in which an apprentice repair person would be given advice about how to take apart and reassemble a small air compressor.[1] The advice would be generated (as needed by the apprentice) by a planning and execution system that would give suggestions about what to do next. The apprentice could ask for advice, using a predefined small vocabulary, which the system would analyze with state-of-the art speech recognition equipment, and then the system would answer, using a speech generation module. The system would also employ a machine-vision system to help keep track of progress and to observe components pointed to by the apprentice. The proposed setup is shown in Fig. 18.1.

Although the project did not last long enough to achieve all of its goals, an early version of a "Computer-Based Consultant" (CBC) system was demonstrated in which the apprentice (played by Georgia Sutherland, one of the project team members) was asked to reassemble an air compressor (a Sears Model 17209, 1/2 HP compressor) that had previously been partially taken apart. Requests and responses by both the system and the apprentice were spoken. Here is a sample of one of the interactions with the system (with my commentary in parentheses):[2]

System: `Please assemble air compressor.`

Apprentice: `How?`

(The apprentice asks for advice, and the system, using Sacerdoti's NOAH system I described previously, creates a hierarchical assembly plan.)

S: `Install pump.`

(The system mentions the first action in the high-level plan NOAH created.)

Figure 18.1. Proposed arrangement for helping an apprentice. (Used with permission of SRI International.)

A: `OK.`
(The apprentice knows how to install the pump and indicates that she does not need further advice about how to do it. At this stage of the CBC project, the system assumed, without visual checking, that the apprentice had completed this first step.)

S: `Install pump brace.`
(The system mentions the second action in the high-level plan.)

A: `How?`
(The apprentice does not know how to achieve this step and asks for advice.)

S: `Connect pump brace to pump.`
(The system mentions the first of the more detailed steps in the high-level action for installing the pump brace.)

A: `OK.`
(The apprentice knows how to achieve this step.)

S: `Connect pump brace to belt-housing frame.`
(The system mentions the next of the more detailed steps in the high-level action for installing the pump brace.)

This type of interaction continues until the task is completed.

The system kept track of the current state of assembly by using the procedural net created by the NOAH system (see p. 176). If the apprentice were to have asked "Why?" at any step, the system would have used the procedural net to construct an answer. The answer might have been something like "you must do x because it

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

is one of the actions involved in doing y," or "you must do x because you must do it first in order to do the next action, z." Also, the apprentice could have asked for help in locating one of the major parts of the air compressor, and the system would have used its laser pointer to show her. Part of the procedural net computed for this example is shown in Fig. 18.2.

The CBC project also provided an opportunity for SRI's NLP group to try out some ideas they were developing about generating and understanding the sentences used in conversations. In the CBC project, the apprentice and the person giving advice are participating in a dialog about a task, namely, the task of working on an air compressor. The structure of the task, as modeled by the procedural network generated by NOAH, provided important pragmatic information useful for sentence understanding. This information was exploited in a system called TDUS (an acronym for Task Dialog Understanding System), which could engage in more complex dialogs than the spoken one just illustrated as it guided an apprentice through an assembly task.[3] TDUS integrated the NOAH planning system with a natural language understanding system (having syntactic, semantic, and pragmatic components) to allow text-based conversations with the apprentice.

I'll use an example taken from a paper about TDUS to illustrate the role that the task structure plays in sentence understanding.[4] Consider the following sentences:

Speaker 1: Why did John take the pump apart?
Speaker 2: *He did it* to fix *it*.

Interpreting the referents of the italicized words in the second sentence is aided by referring to the task context established by the first sentence. "He" refers to John, "did it" refers to the disassembly task, and the second "it" refers to the pump. TDUS makes extensive use of the shifting "context" and goals of the dialog. As the developers of TDUS wrote,[5]

> As a dialog progresses, the participants continually shift their focus of attention and thus form an evolving context against which utterances are produced and interpreted. A speaker provides a hearer with clues of what to look at and how to look at it – what to focus upon, how to focus upon it, and how wide or narrow the focusing should be. We have developed a representation for discourse focusing, procedures for using it in identifying objects referred to by noun phrases, and procedures for detecting and representing shifts in focusing.

(The words "utterance," "speaker," and "hearer" are not to be taken literally. TDUS processed text-based language, not spoken language. In NLP research, these words are often used in a generalized sense to refer to sentences, sentence generators, and sentence receivers, whatever the medium.)

Focus was the main interest of Barbara J. Grosz (1948– ; Fig. 18.3), who continued work on that topic and its role in NLP as a professor at Harvard University. Besides the mechanisms for dealing with contexts, goals, and focus, TDUS contained a grammar, called DIAGRAM,[6] for recognizing many of the syntactic structures of English, means for representing and reasoning about processes and goals, and a framework for describing how different types of knowledge interact as the dialog unfolds.

A demonstration of the CBC system, like the one I described a few paragraphs ago, was given at SRI on April 23, 1975, for J. C. R. Licklider [who had returned to head IPTO in 1973]. Recollecting impressions of his visit, Licklider later said[7]

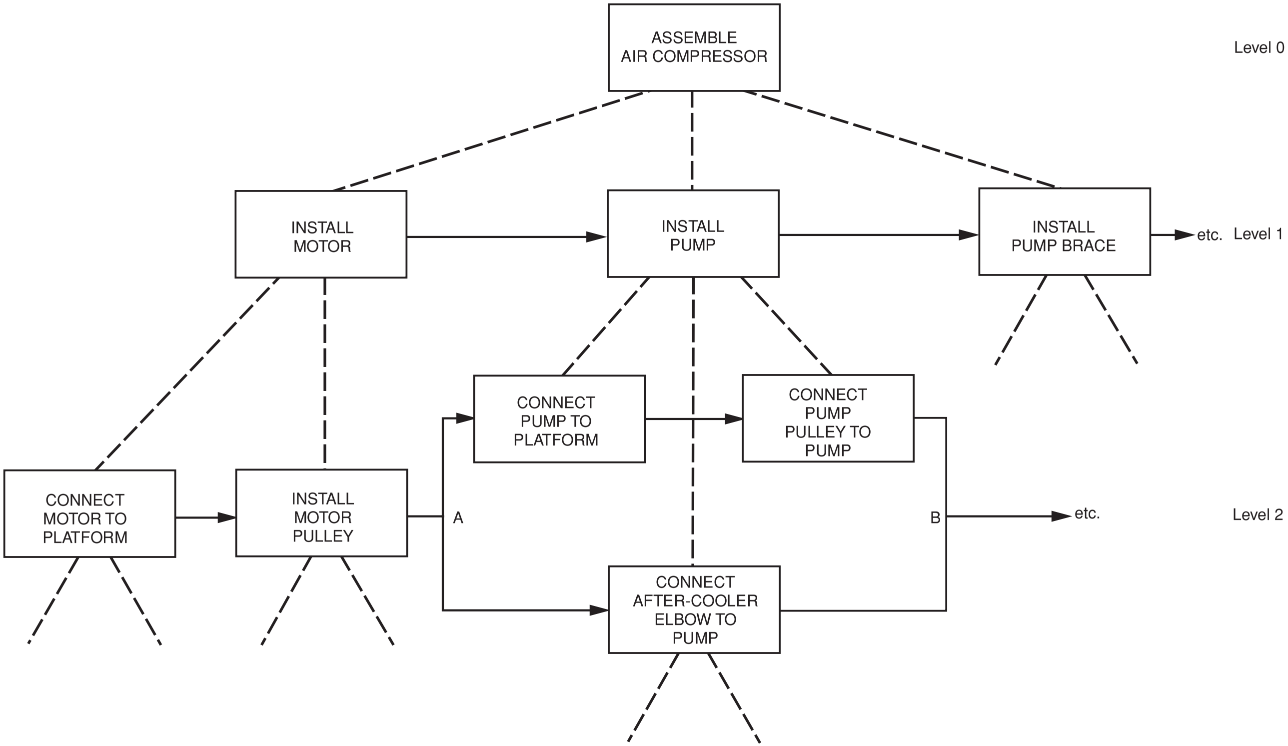

Figure 18.2. Part of a procedural net for assembling an air compressor. (Used with permission of SRI International.)

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

Figure 18.3. Barbara J. Grosz. (Photograph courtesy of photographer Tony Rinaldo.)

The second time I was in DARPA, there were very impressive AI systems dealing with maintenance of equipment. I remember SRI had a program that described how to take a pump apart and put it back together. That's not a terribly complicated device, but it was pretty impressive to see a computer that obviously understood all the parts of the pump and how they worked together.

Because of Licklider's encouragement, we were optimistic about continuing the CBC project and made plans for a system that would diagnose and give advice about repairing a military jeep engine. Unfortunately, one of DARPA IPTO's new program managers, Colonel David Russell, was not buying it. After visiting SRI a few days before Licklider's April 23 visit, Russell sent an e-mail to Licklider saying[8]

I must admit to considerable concern over the SRI program, particularly in light of the management pressures on the AI program. Looking at the projected program plan that Nils has been working on, I see a 2.2M dollar program over the next three years with the aim of developing an experimental CBC for a jeep. . . . I can't see how it can be defended as a near-term application . . .

While it may be difficult, I would suggest that you give serious thought to terminating the CBC program when it completes the air-compressor phase and redirect SRI to more Defense oriented applications or pass their work to NSF. I appreciate that this is heresy, but that is how I saw the situation.

I didn't directly discuss these comments with Nils although I did ask what he would do if the program were terminated. I may have formed a negative view based on an incorrect understanding of the program, and I didn't want to upset the SRI group without your views of the program.

Later that year, Russell replaced Licklider as Director of DARPA IPTO and terminated the CBC project. (Work on TDUS, however, continued under NSF support.) DARPA support for the SRI group was subsequently "redirected" to natural language interfaces to databases (which I'll describe later) and to "image

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

Figure 18.4. Bruce Buchanan (left) and Ted Shortliffe (right). (Photograph courtesy of Ed Feigenbaum.)

understanding" to aid photo interpreters. Some of us chose instead to seek non-DARPA support to work on computer-based consulting systems. Ongoing work at Stanford University on so-called expert systems encouraged us in that direction.

18.2 Expert Systems

18.2.1 MYCIN

Stanford's HEURISTIC DENDRAL project demonstrated the power of endowing computers with expert knowledge about chemistry and spectroscopy. Feigenbaum, Lederberg, and Buchanan, the senior members of the project, believed that a similar approach might work on a medical problem. In the early 1970s Buchanan began talking with Stanley Cohen, Chief of Clinical Pharmacology at Stanford's Medical School, about Cohen's computerized drug interaction warning system called MEDIPHOR. Around the same time, Edward (Ted) Shortliffe (1947– ; Fig. 18.4), a Stanford Medical School student, took a Stanford course on AI and also became an assistant on Cohen's project. Together, Shortliffe, Buchanan, and Cohen conceived the idea of building a computer program that would consult with physicians about bacterial infections and therapy. Shortliffe named the program MYCIN, a common suffix for antibacterial agents. Such a program would need to contain diagnostic and treatment knowledge of experts in infectious diseases.

The first question in developing MYCIN was how to represent expert knowledge. Shortliffe and Buchanan thought that something similar to the "IF–THEN rules" used in DENDRAL would be appropriate. When diagnosing what disease might be causing certain symptoms, as well as in prescribing therapy, physicians appear to be using a kind of IF–THEN reasoning: IF the symptoms are such-and-such, THEN the cause is likely to be so-and-so. The knowledge behind this sort of reasoning is based on experience with cases as well as on scientific knowledge about diseases. It was believed that the IF–THEN knowledge needed by the program could be obtained by interviewing the appropriate medical experts who already thought in those terms.

Interestingly, IF–THEN reasoning about medical matters has a long history. Summarizing part of a book by J. H. Breasted[9] about surgical knowledge contained in an ancient Egyptian papyrus, Robert H. Wilkins wrote "The Edwin Smith Surgical Papyrus, dating from the seventeenth century B.C., is one of the oldest of all known medical papyri."[10] (The papyrus was bought in a Luxor antique shop by Edwin Smith in 1882.) Wilkins goes on to mention several rules from the papyrus, one of which is the following:

Case Thirty

Title: Instructions concerning a sprain in a vertebra of his neck.

Examination: If thou examinest a man having a sprain in a vertebra of his neck, thou shouldst say to him: "look at thy two shoulders and thy breast." When he does so, the seeing possible to him is painful.

Diagnosis: Thou shouldst say concerning him: "One having a sprain in a vertebra of his neck. An ailment which I will treat."

Treatment: Thou shouldst bind it with fresh meat the first day. Now afterward thou shouldst treat [with] ywrw (and) honey every day until he recovers.

Two other experts who joined in the development of the nascent diagnostic and treatment system were Thomas Merigan, Chief of the Infectious Disease Division at Stanford, and Stanton Axline, a physician in that division. In their summary[11] of the history of the project, Buchanan and Shortliffe credit Axline with coming up with the name MYCIN for the program.

The team submitted a successful grant application to the National Institutes of Health in October of 1973. Shortliffe decided to combine his medical studies with work toward a Computer Science Ph.D. based on MYCIN. Since the version of LISP he wanted to use (BBN-LISP, soon to become INTERLISP) was not available at Stanford, he used the SRI AI group's PDP-10 computer.

The IF–THEN rules elicited from the medical experts usually were hedged with uncertainty. Buchanan and Shortliffe mention that "Cohen and Axline used words such as 'suggests' or 'lends credence to' in describing the effect of a set of observations on the corresponding conclusion. It seemed clear that we needed to handle probabilistic statements in our rules . . ."

After wrestling with various ways to use probabilities to qualify MYCIN's IF–THEN rules, Shortliffe finally decided on using the somewhat ad hoc notion of "certainty factors."[12]

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

Here, for example (in both its internal LISP form and its English translation), is one of MYCIN's rules:

```
RULE036
PREMISE: ($AND (SAME CNTXT GRAM GRAMNEG)
               (SAME CNTXTM MORPH ROD)
               (SAME CNTXT AIR ANAEROBIC))
ACTION: (CONCLUDE CNTXT IDENTITY BACTEROIDES TALLY 0.6)

IF: 1) The gram stain of the organism is gramneg, and
    2) The morphology of the organism is rod, and
    3) The aerobicity of the organism is anaerobic

THEN: There is suggestive evidence (0.6) that the identity
            of the organism is bacteroides
```

The 0.6 in this rule is meant to measure the expert's "degree of belief" in or "certainty" about the conclusion. Shortliffe thought that a degree of belief was not the same as a probability assessment because, among other things, he noted that the experts who provided Rule 036 did not necessarily think that the probability of the organism *not* being bacteroides would be 0.4. The original MYCIN system had 200 such rules. By 1978, it had almost 500.

MYCIN's rules were usually evoked in a backward-reasoning fashion. For example, a rule of the form "`IF x1 and x2, THEN y`" would be used if the system's overall goal was to conclude `y`. The use of this rule would lead to the use of rules whose "THEN" parts were either `x1` or `x2`. At the end of a chain of rules, a physician user of the system (or a database) would be asked to supply information about the "IF" part. So, if MYCIN were trying to establish that the identity of an organism was bacteroides, RULE036 would be used and the physician (or database) would be asked if the gram stain of the organism is gramneg and so on.[13]

MYCIN was configured as a "consulting system." That is, it interacted with a physician user who supplied information about a specific patient. The use of rules and rule-chaining allowed the system to provide "explanations" for its reasoning. For example, after a query to the user evoked by Rule 036, if the user asked "Why did you ask whether the morphology of the organism is rod," the system would reply (in English) something like "because I am trying to determine whether the identity of the organism is bacteroides."

So, how did MYCIN do at its primary task of recommending therapy? Shortliffe and colleagues conducted several evaluations in which physicians were asked to compare MYCIN's recommendations with their own for several patients. Their major conclusion was that "Seventy percent of MYCIN's therapies were rated as acceptable by a majority of the evaluators." They also noted, by the way, that "75% is in fact better than the degree of agreement that could generally be achieved by Stanford faculty being assessed under the same criteria."[14]

One of MYCIN's innovations (as contrasted with DENDRAL, say) was that its reasoning process (using the rules) was quite separate from its medical knowledge

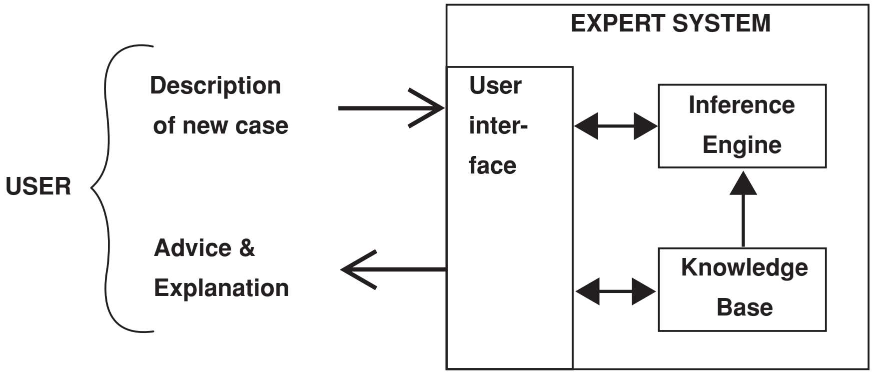

Figure 18.5. The structure of a MYCIN-style expert system.

(the rules themselves). Thus, it became common to divide the program into two parts, namely, the "inference engine" for applying rules and the "knowledge base" of rules. In principle, new rules could be added without having to change the inference engine. This division is shown in Fig. 18.5. This separation suggests that one could construct expert systems for other applications simply by replacing the medical knowledge with some other knowledge base without having to change the inference engine. William van Melle implemented a system he called EMYCIN ("E" for "empty") for doing just that.[15] A system designer along with experts in some field, X, could interact with EMYCIN to produce IF–THEN rules for field X. Using its built-in inference engine, EMYCIN could then use these rules to provide advice to a user of the system during a consultation. EMYCIN was used to build several different expert systems in fields as diverse as tax planning and mechanical structural analysis.

Researchers soon discovered that a minor variation of the certainty factors used by MYCIN and EMYCIN was equivalent to using probabilities instead. This linkage to probability theory implied consequences that neither MYCIN nor EMYCIN could escape. In particular, their reasoning was consistent with probability theory only under some rather restrictive assumptions about how rules were used. As Russell and Norvig point out, if these assumptions aren't met "certainty factors could yield disastrously incorrect degrees of belief through overcounting of evidence. As rule sets became larger, undesirable interactions between rules became more common, and practitioners found that the certainty factors of many other rules had to be 'tweaked' when more rules were added."[16] Modern methods use more sophisticated probabilistic techniques, as we shall see in a later chapter.

Even so, the success of MYCIN and the various EMYCIN programs led to the development of many more expert systems, some based on EMYCIN and some using their own specific approaches. As Allen Newell wrote in his introduction to a book by Buchanan and Shortliffe, "MYCIN is the original expert system that made it evident to all the rest of the world that a new niche had opened up. . . . MYCIN epitomized the new path that had been created. Thus, gathering together the full record of this system and the internal history of its development serves to record an important event in the history of AI."[17]

18.2.2 PROSPECTOR

Inspired by Shortliffe's work with MYCIN, some of us at SRI began investigating nonmedical applications of expert systems. One area we considered was "integrated pest management" in which knowledge about crops and their insect pests could be used to mitigate the effects of insect predation with minimal use of chemical insecticides. Although proposals were written and some interest was shown by scientists in the U.S. Department of Agriculture and at the Environmental Protection Agency, the idea was abandoned when the proposals went unfunded.

Peter Hart and Richard Duda eventually focused on systems for providing advice to explorationists about possible "hard-rock" mineral deposits.[18] Hart had some early discussions with John Harbaugh, a petroleum engineering professor at Stanford, and with Alan Campbell, one of Harbaugh's graduate students. (Alan Campbell was the son of the late Neal Campbell, a world-famous explorationist who had discovered what was possibly the largest lead–zinc deposit in the world. Alan spent much of his youth in mining camps.) Through Campbell, Hart and Duda met Charles Park, the former Dean of Stanford's School of Earth Sciences and an authority on hard-rock mineral deposits. Park helped Hart and Duda codify knowledge about lead–zinc deposits in the form of IF–THEN rules. Further work with Marco Einaudi, a professor in Stanford's Department of Economic Geology, led to additional rules and rule-organizing ideas. Ultimately the U.S. Geological Survey provided funding for the development of what became the PROSPECTOR expert system for consultation about mineral deposits.[19]

A large group of people participated in the design and writing of the PROSPECTOR program. Duda and Hart led the effort. I joined the project sometime after work had begun and after hearing from DARPA that the CBC project was not going to be continued. Other contributors were John Gaschnig (1950–1982), Kurt Konolige, René Reboh, John Reiter, Tore Risch, and Georgia Sutherland. MYCIN was a dominant influence on the technology being developed – "primarily through its use of rules to represent judgmental knowledge, and its inclusion of formal mechanisms for handling uncertainty."[20] Other important influences came from another medical diagnosis system, INTERNIST-1, which I'll describe shortly. These were its use of taxonomic information and its ability to handle volunteered (rather than only queried) information.

PROSPECTOR used rules to make inferences and to guide the consultation process. Two examples of these rules are

Rule 3: "Barite overlying sulfides suggests the possible presence of a massive sulfide deposit."

and

Rule 22: "Rocks with crystal-shaped cavities suggest the presence of sulfides."

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

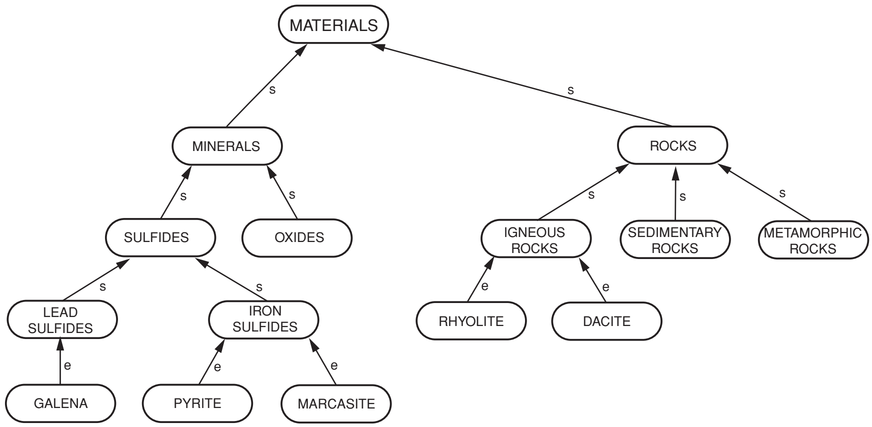

Figure 18.6. A partial geologic taxonomy. (Used with permission of SRI International.)

The rules were encoded as "partitioned semantic networks" – a format originated by Gary Hendrix (1948–) in his University of Texas Ph.D. thesis for use in representing knowledge needed by natural language processing systems.[21] Semantic networks were also used to represent the taxonomic knowledge used by PROSPECTOR. An example of such a network is shown in Fig. 18.6. The rules could be linked together in what was called an "inference network." A simplified example for reasoning about a Kuroko-type massive sulfide deposit is shown in Fig. 18.7. Note how Rule 22 helps to establish one of the premises for Rule 3. Note also that the taxonomy is used to infer the presence of sulfides when galena, sphalerite, or chalcopyrite is known to be present.

Inferences from rule premises to rule conclusions in the network depended on probabilities and Bayes's rule – not on ad hoc numbers such as "certainty factors." The geological experts were asked to quantify their uncertainty about a rule by giving the designers two numbers. One is the factor by which the odds favoring the conclusion would be increased if the premises were true. The other is the factor by which the odds favoring the conclusion would be decreased if the premises were false. Bayes's rule was used in association with these numbers to derive the probability of the conclusion given the probabilities of the premises.[22] PROSPECTOR's inference methods, even though they were an improvement over those of MYCIN, gave probabilistically valid results only for certain kinds of inference-net structures. As Glenn Shafer and Judea Pearl explain, "Probabilities could not simply tag along as numbers attached to IF–THEN rules. The results of probability calculations would be sensible only if these calculations followed principles from probability theory."[23] Modern expert systems use the more general framework of Bayesian networks, which will be described later.

The usual format for a PROSPECTOR consultation involved a session with a geologist interested in evaluating a certain site. The geologist might volunteer some information, which would evoke some of PROSPECTOR's rules. The system then calculated what additional information would be most effective in altering the

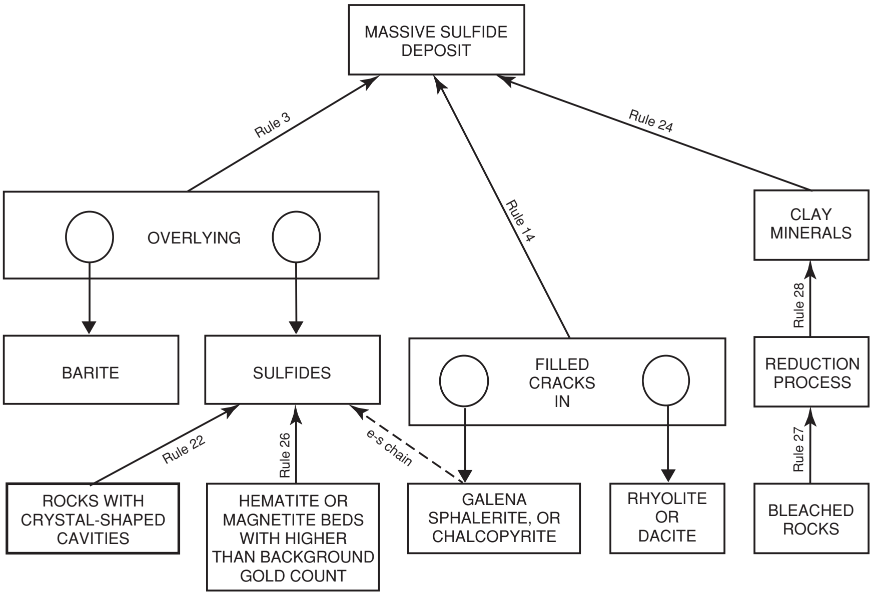

Figure 18.7. Simplified version of a PROSPECTOR inference network. (Used with permission of SRI International.)

probability of whatever it was the geologist wanted to find out. PROSPECTOR would then ask a question to elicit that information (and its probability). Throughout the process, the user could volunteer additional information at any time.

Because PROSPECTOR could use volunteered information, a run of the program need not be part of a question-and-answer consultation session. Instead, a user could input a whole set of data about "findings in the field" to PROSPECTOR, which would then draw its conclusions. These findings could be from a database or, perhaps more usefully, from a map that indicated contours of regions in which various kinds of minerals were found to be present. (Kurt Konolige of SRI joined the PROSPECTOR team around this time and wrote a program that allowed PROSPECTOR to use map data as an input.)

The most dramatic instance of PROSPECTOR's use of map data occurred when it successfully identified the location of a porphyry molybdenum deposit at Mount Tolman in the state of Washington.[24] Results of previous exploration of the Mount Tolman site were used to produce maps outlining important geological data relevant to potential molybdenum deposits. PROSPECTOR processed these maps using rules obtained primarily from Victor F. Hollister, an expert on porphyry molybdenum deposits, and Alan Campbell. The result of the processing was another map indicating the relative "favorability" of a mineral deposit. Computer displays of some of the input maps are shown in Fig. 18.8. I won't explain the geological details of what these maps depict, but they represent the kind of data thought to be important by experts such as Campbell and Hollister.

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

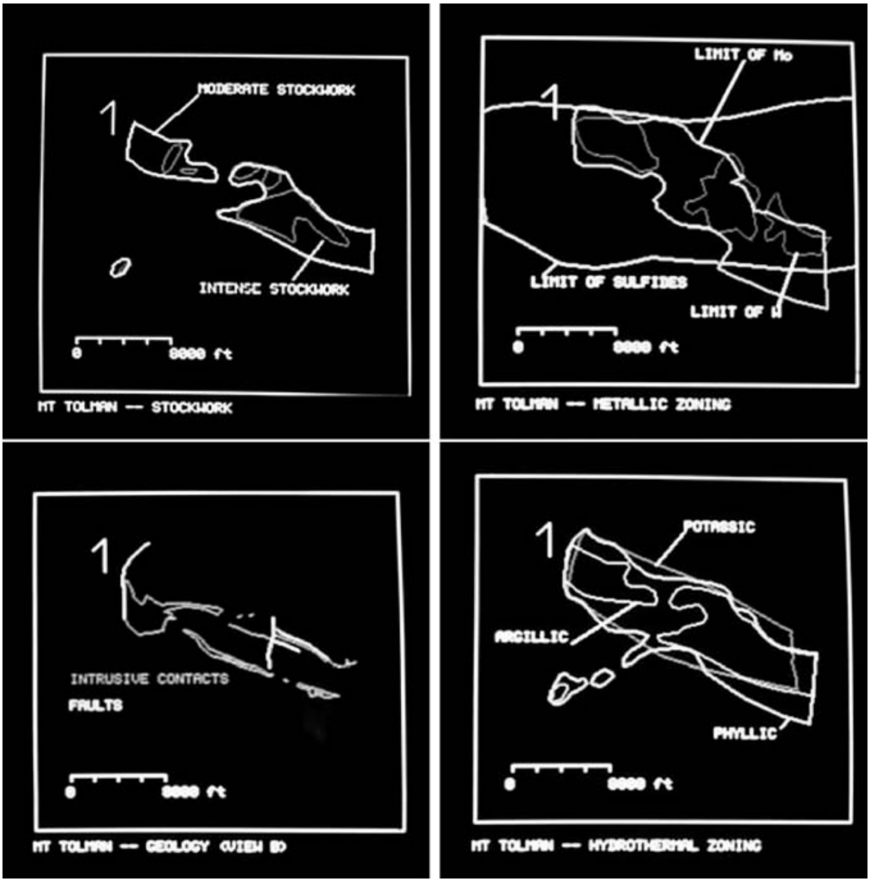

Figure 18.8. Some of the Mount Tolman input maps. (Photographs courtesy of Richard Duda.)

From data of this sort, PROSPECTOR produced favorability maps, one of which is shown in Fig. 18.9. The scale on the right of the map (when rendered in color) indicates favorability from +5 (highly favorable) through −5 (highly unfavorable). Based on previous extensive drilling in the largest of the favorable areas, a mining company had planned an open-pit mine there (outlined by the contour labeled "proposed pit").

One must be careful in evaluating this result. It is not the case that PROSPECTOR discovered an ore deposit in a site previously unexplored. As was pointed out in a letter to the editor of the journal *Artificial Intelligence*,[25]

A large mining company had already found a molybdenum ore body by drilling over 200 exploration holes in one region . . . and we knew that they intended to do further drilling for their own information.

. . .

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

Figure 18.9. Favorability map. (Photograph courtesy of Richard Duda.)

[This further drilling] showed a remarkable congruence with PROSPECTOR's favorability map, including both verification of PROSPECTOR's prediction of a large, previously unknown region of ore-grade mineralization, and verification of PROSPECTOR's predictions for the barren areas.

...

Unfortunately, prolonged depressed economic conditions in the minerals industry have made this area unprofitable to mine . . . Thus, PROSPECTOR's success to date has been scientific rather than economic.

Readers interested in more details should see the *Science* article previously cited and a summarizing final report on the PROSPECTOR project.[26]

The computer code for PROSPECTOR was delivered to the U.S. Geological Survey where Richard B. McCammon developed a successor system he called PROSPECTOR II. Summarizing his system, McCammon wrote[27]

PROSPECTOR II, the successor to PROSPECTOR, was developed at the US Geological Survey. Currently, the knowledge base contains 86 deposit models and information on more than 140 mineral deposits. Within minutes, the geologist can enter the observed data for an area, select the types of deposit models to be evaluated, receive advice on those models that best match the observed data, and, for a particular model, find out which of the data can be explained, which of the data are unexplained, and which critical attributes of the model are not observed in the data.

18.2.3 *Other Expert Systems*

Several other expert systems followed the MYCIN and PROSPECTOR work. Some, like MYCIN, were for medical diagnosis and therapy.[28] Of these, I'll mention the INTERNIST-1 program by computer scientists Randolph A. Miller and Harry E. Pople and physician Jack D. Myers at the University of Pittsburgh and the CASNET (Causal-ASsociational NETwork) program by Casimir A. Kulikowski and Sholom M. Weiss of Rutgers University.

The INTERNIST-1 series of diagnosis programs contained expertise about internal medicine.[29] Part of this knowledge was represented in a kind of semantic network

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

or taxonomy of disease states (called a *nosology* in medicine). In an article in the *New England Journal of Medicine*, Miller, Pople, and Myers state that the performance of INTERNIST-1 "on a series of 19 clinicopathological exercises (Case Records of the Massachusetts General Hospital) published in the Journal appeared qualitatively similar to that of the hospital clinicians but inferior to that of the case discussants." However, they concluded that "the present form of the program is not [yet] sufficiently reliable for clinical applications."[30] Later, much of the diagnostic knowledge assembled in INTERNIST-1 was repackaged in QMR (Quick Medical Reference), a diagnostic decision support system for internists.[31] (It has since been discontinued by its eventual purchaser First DataBank.)

CASNET also used networks.[32] In those, "inference rules" linked observations, patho-physiological states, diagnostic states, and treatment states. Their primary application was to the glaucomas, for which they had good physical models on which the inference rules could be based.

At Carnegie Mellon University, John McDermott (1942–) helped in the development of a rule-based system called XCON (for eXpert CONfigurer) to assist in the ordering and configuring of Digital Equipment Corporation's VAX computer systems. XCON grew out of an earlier system by McDermott called R1.[33] R1 and XCON were written in a special rule-processing language called OPS5, one of the OPS family of languages developed by Charles Forgy (1949–) at CMU.[34] (OPS is said to be an acronym for Official Production System.) The OPS languages used Forgy's "Rete" algorithm for efficiently stringing IF–THEN rules together.[35] XCON first went into use in 1980 in DEC's plant in Salem, New Hampshire.[36]

The problem with how to deal with uncertain information was avoided in XCON because it almost never encountered a configuration issue that it did not have enough certain knowledge to handle. By 1989, according to a paper about XCON and related configuration systems at DEC,[37] these systems had a total of about 17,500 rules. The paper went on to say that

> . . . overall the net return to Digital is estimated to be in excess of $40 million per year.
>
> The use of the configuration systems insures that complete, consistently configured systems are shipped to the customer. Incomplete orders do not get through the process. In addition, XCON generates configurations which optimize system performance, so customers consistently get the best view of our products. Before the configuration systems, we would often ship the same parts configured differently.

In addition to XCON and its DEC siblings, several expert systems were built and put in use by companies and research laboratories during the 1980s. In 1983, General Electric developed the Diesel Electric Locomotive Troubleshooting Aid (DELTA), a prototype system to assist railroad personnel in the maintenance of General Electric's diesel-electric locomotives. The developers stated that it "can diagnose multiple problems with the locomotive and can suggest repair procedures to maintenance personnel." It had 530 rules "partially representing the knowledge of a Senior Field Service Engineer."[38]

Another example is JETA (Jet Engine Troubleshooting Assistant), developed by engineers at the National Research Council in Canada. According to a paper about JETA, it "has been applied to troubleshoot the General Electric J85-CAN-

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

15 jet engine that powers the CF-5 trainer fighters used by the Canadian Air Force."[39] Knowledge about jet engines and their possible faults and symptoms are encoded in frames. Rules are used solely for "specific control functions embedded in a frame and for asynchronous user input."

An expert system called CCH-ES for credit analysis was put in use at the Credit Clearing House (CCH) division of Dun & Bradstreet (D&B) in July 1989. It contained approximately 800 rules and could handle online transactions when CCH customers called in for service or when analysts wanted to review cases. Batch cases were run when there were updates in the relevant databases. According to a paper about the system, "Analyst agreement with CCH-ES continues to be at approximately 98.5 percent on an ongoing basis. . . . [It] has been a major success at D&B. It has provided CCH with an automated credit analyst expert system that can provide expert-level credit analysis decisions consistently and at a high-quality level. Customers have uniformly praised the system."[40]

More expert systems are described in the book *The Rise of the Expert Company*.[41] In an appendix to that book, Paul Harmon lists over 130 expert systems in use during the mid- to late 1980s, including

- *Grain Marketing Advisor* for helping farmers choose marketing or storage strategies for their grain crops,
- *ACE* for helping telephone operating companies reduce the incidence of phone cable failures,
- *IDEA* for helping technicians diagnose trouble situations in the Infotron IS4000 Local Area Network,
- *Diag 8100* for helping with the diagnosis of problems and failures in IBM 8100 computers at the Travelers Corporation,
- *Intelligent Peripheral Troubleshooter* for helping to troubleshoot Hewlett-Packard disk drives,
- *SNAP* for helping shoppers at Infomart (a Dallas computer store) assess their personal computer needs,
- *Pile Selection* for helping designers at the Kajima Construction Company select piling material to be used in the foundations of buildings,
- *ExperTAX* for helping to evaluate the application of new U.S. tax laws for clients of Coopers and Lybrand, and
- *Dipmeter Advisor* for helping in the analysis of geological formations encountered in oil-well drilling.

18.2.4 *Expert Companies*

New companies and divisions of established companies were started to develop and field these applications. The first of these was Teknowledge, organized by a group of Stanford faculty and researchers to market expert systems and to consult about expert systems. Teknowledge used EMYCIN as its basic technology. Another was Syntelligence, founded by Peter Hart and Richard Duda (along with some of the PROSPECTOR researchers) to market expert systems for insurance underwriting and loan credit analysis. At Syntelligence, expert systems were written in the

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press

SYNTEL language, developed by René Reboh and Tore Risch and based on ideas from PROSPECTOR. After leaving CMU, Charles Forgy founded Production Systems Technologies in 1983 "to develop and market state of the art rule-based tools."[42] Among other companies formed during this period were Aion Corporation, Helix Expert System, Ltd., Exsys, Inc., Inference Corporation, and IntelliCorp.[43] Because it was not too difficult for clients who wanted expert systems to develop their own versions (which were able to run on low-cost workstations and personal computers), many of the expert systems companies ceased to exist, were bought by larger companies, or had to reorient their businesses to provide additional or related services.

After the flurry of excitement over expert systems died down a bit in the 1980s and 1990s, some developers concentrated on systems for acquiring and deploying "business rules." According to an organization called the Business Rules Group, a business rule is "a statement that defines or constrains some aspect of the business. It's intended to assert business structure, or to control or influence the behavior of the business."[44] For example, a business rule might state "when our widget inventory is below 200, notify widget production." Business rules take the form of IF–THEN statements, just like expert-system rules. In business applications, expert-system inference engines metamorphosed into business rule engines (BREs). They are used either to answer questions about business practices or to take actions such as placing orders or sending alerts.[45] Some of the people who had been involved in providing expert systems software switched to business-rule software. For example, in 2002, Charles Forgy founded RulesPower, Inc., whose business rules management systems (BRMSs) used later versions of the Rete algorithm. (In 2005, RulesPower sold some of its assets to Fair Isaac Corporation, an analytics and decision management technology company, which has since changed its name to FICO.)

Notes

1. See Nils J. Nilsson *et al.*, "Plan for a Computer-Based Consultant System," SRI AI Center, Technical Note 94, May 1974. (Available online at http://www.ai.sri.com/pubs/files/1298.pdf.) **[224]**
2. From Peter E. Hart, "Progress on a Computer Based Consultant," SRI AI Center Technical Note 99, p. 23, January 1975. (Available online at http://www.ai.sri.com/pubs/files/1389.pdf.) **[224]**
3. Ann E. Robinson, Douglas E. Appelt, Barbara J. Grosz, Gary G. Hendrix, and Jane J. Robinson, "Interpreting Natural-Language Utterances in Dialogs About Tasks," SRI AI Center Technical Note 210, March 15, 1980. (Available online at http://www.ai.sri.com/pubs/files/709.pdf.) **[226]**
4. *Ibid*, p. 11. **[226]**
5. *Ibid*, p. 11. **[226]**
6. Jane J. Robinson, "DIAGRAM," SRI AI Center Technical Note No. 205, 1980; available online as SRI AI Center Technical Note 205, February 1980, at http://www.ai.sri.com/pubs/files/712.pdf. **[226]**
7. J. C. R. Licklider, "The Early Years: Founding IPTO," in Thomas C. Bartee (ed.), *Expert Systems and Artificial Intelligence: Applications and Management*, p. 223, Indianapolis, IN: Howard W. Sams & Co., 1988. **[226]**
8. A copy of this e-mail is in my files. **[228]**

9. J. H. Breasted, *The Edwin Smith Surgical Papyrus*, two volumes, Chicago: University of Chicago Press, 1980. **[230]**
10. See http://www.neurosurgery.org/cybermuseum/pre20th/epapyrus.html for a copy of the Wilkins article. **[230]**
11. Bruce G. Buchanan and Edward H. Shortliffe (eds.), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, MA: Addison-Wesley, 1984. The book is now out of print but is available online at http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/RuleBasedExpertSystems. Shortliffe's dissertation has been reprinted as a book: Edward H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*, New York: Elsevier, 1976. **[230]**
12. Others too attempted to use ideas not strictly based on probability theory. Among these were Arthur Dempster and Glenn Shafer (see Glenn Shafer, *A Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976) and Lotfi Zadeh, who developed "fuzzy logic" (see, as just one of many sources, Lotfi A. Zadeh, "A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts," *International Journal of Man-Machine Studies*, Vol. 8, pp. 249–291, 1976, available online at http://www-bisc.cs.berkeley.edu/ZadehFA-1976.pdf.) I'll mention these alternatives later in the book. **[230]**
13. For a full description of how MYCIN's rules were acquired and used see Bruce G. Buchanan and Edward H. Shortliffe, *op. cit.* **[231]**
14. Bruce G. Buchanan and Edward H. Shortliffe, *op. cit.*, Chapters 30 and 31. **[231]**
15. EMYCIN is described in Bruce G. Buchanan and Edward H. Shortliffe, *op. cit.*, Chapter 15. EMYCIN was the subject of van Melle's Ph.D. dissertation: William van Melle, "A Domain-Independent System That Aids in Constructing Knowledge-Based Consultation Programs," Stanford University Computer Science Department; see also Stanford Report Nos. STAN-CS-80-820 and HPP-80-22, 1980. **[232]**
16. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, second edition, p. 525, Upper Saddle River, NJ: Prentice Hall, 2003. **[232]**
17. Bruce G. Buchanan and Edward H. Shortliffe, *op. cit.* **[233]**
18. Economic geologists distinguish mineral deposits from ore deposits. An *ore* is a mineral that can be *profitably* extracted. Hard-rock minerals include copper, lead, zinc, and so on, but not hydrocarbons. **[233]**
19. PROSPECTOR was first described in Peter E. Hart, "Progress on a Computer-Based Consultant," *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 831–841, 1975. **[233]**
20. Richard O. Duda *et al.*, "Semantic Network Representations in Rule-Based Inference Systems," in D. A. Waterman and Frederick Hayes-Roth (eds.), *Pattern-Directed Inference Systems*, Orlando, FL: Academic Press, Inc., 1978. Available online at http://www.ai.sri.com/pubs/files/751.pdf. **[233]**
21. Gary G. Hendrix, "Partitioned Networks for the Mathematical Modeling of Natural Language Semantics," Ph.D. thesis, University of Texas Computer Science Department, 1975. For a short paper, see Gary G. Hendrix, "Expanding the Utility of Semantic Networks Through Partitioning," *Proceedings of the Fourth International Conference on Artificial Intelligence*, pp. 115–121, 1975. This paper also appeared as SRI AI Center Technical Note 105 and is available online at http://www.ai.sri.com/pubs/files/1380.pdf. **[234]**
22. For a description of PROSPECTOR's inference methods see Richard O. Duda, Peter E. Hart, and Nils J. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems," in *Proceedings of the AFIPS National Computer Conference*, Vol. 45, pp. 1075–1082, 1976. Reprinted in G. Shafer and J. Pearl (eds.), *Readings in Uncertain Reasoning*,

pp. 274–281, San Francisco: Morgan Kaufmann Publishers, 1990. A version appears as SRI AI Center Technical Note 124 and is available online at http://www.ai.sri.com/pubs/files/755.pdf. **[234]**

23. Glenn Shafer and Judea Pearl (eds.), *Readings in Uncertain Reasoning*, San Francisco: Morgan Kaufmann Publishers, 1990. The book is no longer in print, but some of the chapters are available online at http://www.glennshafer.com/books/rur.html. **[234]**
24. Alan N. Campbell, Victor F. Hollister, Richard O. Duda, and Peter E. Hart, "Recognition of a Hidden Mineral Deposit by an Artificial Intelligence Program," *Science*, Vol. 217, No. 4563, pp. 927–929, September 3, 1982. **[235]**
25. Richard O. Duda, Peter E. Hart, and René Reboh, letter to the editor, *Artificial Intelligence*, Vol. 26, pp. 359–360, 1985. **[236]**
26. Richard O. Duda, "The PROSPECTOR System for Mineral Exploration," Final Report prepared for the Office of Resource Analysis, U.S. Geological Survey, Reston, VA 22090, April 1980. **[237]**
27. Richard B. McCammon, "PROSPECTOR II – An Expert System for Mineral Deposit Models," *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, Vol. 33, No. 6, pp. 267A–267A(1), September 1996. See also Richard B. McCammon, "PROSPECTOR II," in H. J. Antonisse, J. W. Benoit, and B. G. Silverman (eds.), *Proceedings of the Annual AI Systems in Government Conference*, pp. 88–92, March 1989, Washington, DC. **[237]**
28. See Peter Szolovits (ed.), *Artificial Intelligence in Medicine*, Boulder, CO: Westview Press, 1982. Available online at http://groups.csail.mit.edu/medg/ftp/psz/AIM82/ch0.html. **[237]**
29. Harry E. Pople Jr., "Heuristic Methods for Imposing Structure on Ill-Structured Problems: The Structuring of Medical Diagnostics," Chapter 5 in Peter Szolovits (ed.), *Artificial Intelligence in Medicine*, Boulder, CO: Westview Press, 1982. Available online at http://groups.csail.mit.edu/medg/ftp/psz/AIM82/ch5.html. **[237]**
30. Randolph A. Miller *et al.*, "INTERNIST-1: An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine," *New England Journal of Medicine*, Vol. 307, pp. 468–76, August 19, 1982. **[238]**
31. Randolph A. Miller *et al.*, "The INTERNIST-1/Quick Medical Reference Project – Status Report," *The Western Journal of Medicine*, Vol. 145, No. 6, pp. 816–822, 1986. Available online at http://www.pubmedcentral.nih.gov/picrender.fcgi?artid=1307155&blobtype=pdf. **[238]**
32. Casimir A. Kulikowski and Sholom M. Weiss, "Representation of Expert Knowledge for Consultation: The CASNET and EXPERT Projects," Chapter 2 in P. Szolovits (ed.), *Artificial Intelligence in Medicine*, Boulder, CO: Westview Press, 1982. Available online at http://groups.csail.mit.edu/medg/ftp/psz/AIM82/ch2.html. **[238]**
33. John McDermott, "R1: A Rule-Based Configurer of Computer Systems," *Artificial Intelligence*, Vol. 19, No. 1, pp. 39–88, 1980. **[238]**
34. Charles Forgy, "OPS5 User's Manual," Technical Report CMU-CS-81-135, Carnegie Mellon University, 1981. See also Lee Brownston *et al.*, *Programming Expert Systems in OPS5*, Reading, MA: Addison-Wesley, 1985. **[238]**
35. Charles Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence*, Vol. 19, pp. 17–37, 1982. **[238]**
36. See http://en.wikipedia.org/wiki/Xcon. **[238]**
37. Virginia E. Barker and Dennis E. O'Connor, "Expert Systems for Configuration at Digital: XCON and Beyond," *Communications of the ACM*, Vol. 32, No. 3, pp. 298–318, March 1989. **[238]**

38. Piero P. Bonissone and H. E. Johnson Jr., " DELTA: An Expert System for Diesel Electric Locomotive Repair," *Proceedings of the Joint Services Workshop on Artificial Intelligence in Maintenance*, Boulder, CO, October 4–6, 1983, AD-A145349, pp. 397–413, June 1984 (Defense Technical Information Center Accession Number ADA145349.) **[238]**
39. Phillippe L. Davidson *et al.*, "Intelligent Troubleshooting of Complex Machinery," *Proceedings of the Third International Conference on Industrial Engineering Applications of Artificial Intelligence Expert Systems*, pp. 16–22, Charleston, South Carolina, USA, July 16–18, 1990. See also M. Halasz *et al.*, "JETA: A Knowledge-Based Approach to Aircraft Gas Turbine Engine Maintenance," *Journal of Applied Intelligence*, Vol. 2, pp. 25–46, 1992. **[239]**
40. Roger Jambor *et al.*, "The Credit Clearing House Expert System," *IAAI-91 Proceedings*, pp. 255–269, 1991. **[239]**
41. Edward Feigenbaum, Pamela McCorduck, and H. Penny Nii, *The Rise of the Expert Company: How Visionary Companies Are Using Artificial Intelligence to Achieve Higher Productivity and Profits*, New York: Times Books, 1988. **[239]**
42. http://www.pst.com/. **[240]**
43. Harmon's appendix, just cited, lists several companies as does http://dmoz.org/Computers/Artificial_Intelligence/Companies/. **[240]**
44. From http://www.businessrulesgroup.org/defnbrg.shtml. **[240]**
45. I thank Paul Harmon, now Executive Editor of Business Process Trends (www.bptrends.com), for enlightening me about business rules. **[240]**

https://doi.org/10.1017/CBO9780511819346.022 Published online by Cambridge University Press