PART SIX

Explaining the Reasoning

17

Explanation as a Topic of AI Research

In describing MYCIN's design considerations in Chapter 3, we pointed out that an ability of the program to explain its reasoning and defend its advice was an early major performance goal. It would be misleading, however, to suggest that explanation was a primary focus in the original conception. As was true for many elements of the system, the concept of system transparency evolved gradually during the early years. In reflecting on that period, we now find it impossible to recall exactly when the idea was first articulated. The SCHOLAR program (Carbonell, 1970a) was our working model of an interactive system, and we were trying to develop ways to use that model for both training and consultation. Thus, with hindsight, we can say that the issue of making knowledge understandable was in our model, although it was not explicitly recognized at first as a research issue of importance.

17.1 The Early Explanation Work

When the first journal article on MYCIN appeared in 1973 (Shortliffe et al., 1973), it included examples of the program's first rudimentary explanation capabilities. The basic representation and control strategies were relatively well developed at that time, and it was therefore true that any time the program asked a question some domain rule under consideration had generated the inquiry. To aid with system debugging, Shortliffe had added a RULE command that asked MYCIN to display (in LISP) the rule currently under consideration. At the weekly research meetings it was acknowledged that if the rules were displayed in English, rather than in LISP, they would provide a partial justification of the question for the user and thereby be useful to a physician obtaining a consultation. We then devised the translation mechanism (described in Chapter 5), assigning the TRANS

property to all clinical parameters, predicate functions, and other key data structures used in rules. Thus, when a user typed "RULE" in response to a question from MYCIN, a translation of the current rule was displayed as an explanation. This was the extent of MYCIN's explanation capability when the 1973 paper was prepared.

At approximately the same time as that first article appeared, Gorry published a paper that influenced us greatly (Gorry, 1973). In retrospect, we believe that this is a landmark essay in the evolution of medical AI. In it he reviewed the experience of the M.I.T. group in developing a program that used decision analysis techniques to give advice regarding the diagnosis of acute renal failure (Gorry et al., 1973). Despite the successful decision-making performance of that program, he was concerned by its obvious limitations (p. 50):

> Decision analysis is a useful tool when the problem has been reduced to a small, well-defined task of action selection. [However,] it cannot be the sole basis of a program to assist clinicians in an area such as renal disease.

He proceeded to describe the M.I.T. group's nascent work on an AI system that used "experimental knowledge" as the basis for understanding renal diseases[1] and expressed excitement about the potential of the symbolic reasoning techniques he had recently discovered (p. 50):

> The new technology [AI] . . . has greatly facilitated the development [of the prototype system] and it seems likely that a much improved program can be implemented. The real question is whether sufficient improvement can be realized to make the program useful. At present, we cannot answer the question, but I can indicate the chief problem areas to be explored: [concept identification, language development, and explanation].

We will not dwell here on his discussion of the first two items, but regarding the third (p. 51):

> ..If experts are to use and improve the program directly, then it must be able to explain the reasons for its actions. Furthermore, this explanation must be in terms that the physician can understand. The steps in a deduction and the facts employed must be identified for the expert so that he can correct one or more of them if necessary. As a corollary, the user must be able to find out easily what the program knows about a particular subject.

Gorry's discussion immediately struck a sympathetic chord for us in our own work. The need for explanation to provide transparency and to encourage acceptance by physicians seemed immediately intuitive, not only for expert system builders (as Gorry discussed) but also for the eventual

[1]This program later became the Present Illness Program (Pauker et al., 1976).

end-users of consultation systems.[2] Our early RULE command, however, did not meet the criteria for explanation outlined by Gorry above.

During the next two years, the development of explanation facilities for MYCIN became a major focus of the research effort. Randy Davis had joined the project by this time, and his work on the TEIRESIAS program, which would become his thesis, started by expanding the simple RULE command and language translation features that Shortliffe had developed. Davis changed the RULE command to WHY and implemented a history tree (see Chapter 18) that enabled the user to examine the entire reasoning chain upward to the topmost goal by asking WHY several times in succession. He also developed the HOW feature, which permitted the user to descend alternate branches of the reasoning network. By the time the second journal article appeared in 1975 (Shortliffe et al., 1975), explanation and early knowledge acquisition work were the major topics of the exposition.[3]

In addition to the RULE command, Shortliffe developed a scheme enabling the user to ask free-text questions at the end of a session after MYCIN had given its advice. He was influenced in this work by Dr. Ken Colby, then at Stanford and actively involved in the development of the PARRY program (Colby et al., 1974). Shortliffe was not interested in undertaking cutting-edge research in natural language understanding (he had taken Roger Schank's course at Stanford in computational linguistics and realized it would be unrealistic to tackle the problem exhaustively for a limited portion of his own dissertation work). He was therefore convinced by Colby's suggestion to exploit existing methods, such as keyword search, and to take advantage of the limited vocabulary used in the domain of infectious diseases. The resulting early version of MYCIN's question-answering system was described in a chapter of his dissertation (Shortliffe, 1974).

When Carli Scott first joined the project, she was completing a master's degree in computer science and needed a project to satisfy her final requirements. She was assigned the task of refining and expanding the question-answering (QA) capability in the program. Not only did this work complete her M.S. requirements, but she continued to devote much of her time to explanation during her next few years with the project. She was assisted in this work by Bill Clancey, then a Ph.D. candidate in computer science, who joined us at about the same time. MYCIN's explanation capability was tied to its rule-based representation scheme, so Clancey was particularly interested in how the therapy algorithm might be transferred from LISP code into rules so that it could be made accessible to the explanation routines. His work in this area is the subject of Chapter 6 in this volume.

---

[2]Almost ten years later we undertook a formal study (described in Chapter 34) that confirmed this early intuition. A survey of 200 physicians revealed that high-quality explanation capabilities were the most important requirement for an acceptable clinical consultation system.

[3]This simple model of explanations still has considerable appeal. See Clark and McCabe (1982) for a discussion of implementing WHY and HOW in PROLOG, for example.

By late 1976 the explanation features of the system had become highly polished, and Scott, Clancey, Davis, and Shortliffe collaborated on a paper that appeared in the *American Journal of Computational Linguistics* in 1977. That paper is included here as Chapter 18. It describes MYCIN's explanation capabilities in some detail. Although most of the early work described in that chapter stressed the need to provide explanations to *users,* we have also seen the value such capabilities have for *system builders.* As mentioned in Chapters 9 and 20, system builders—both experts and knowledge engineers—find explanations to be valuable debugging aids. The features described in Chapter 18 were incorporated into EMYCIN and exist there relatively unchanged to the present.

17.1.1 Explaining the Pharmacokinetic Dosing Model

By the mid-1970s much of the project time was being spent on knowledge base refinement and enhancement. Because we needed assistance from someone with a good knowledge of the antimicrobial agents in use, we sought the involvement of a clinical pharmacist. Sharon Bennett, a recent pharmacy graduate who had taken a clinical internship at the Palo Alto Veterans Administration Hospital affiliated with Stanford, joined the project and played a key role in knowledge base development during the mid- to late-1970s. Among the innovations she brought to the group was an eagerness to heighten MYCIN's utility by making it an expert at dosage adjustment as well as drug selection. She and Carli Scott worked together closely to identify the aspects of pharmacokinetic modeling that could be captured in rules and to identify the elements that were so mathematical in nature that they required encoding in special-purpose functions. By this time, however, the need for explanation capabilities had become so obvious to the project's members that even this specialized code was adapted so that explanations could be provided. A paper describing the features, including a brief discussion of explanation of dosing, was prepared for the *American Journal of Hospital Pharmacy* and is included here as Chapter 19. We include the paper here not only because it demonstrates the special-purpose explanation features that were developed, but also because it shows the way in which mathematical modeling techniques were integrated into a large system that was otherwise dependent on AI representation methods.

17.2 Recent Research in Explanation

Even after research on MYCIN terminated, the development of high-performance explanation capabilities for expert systems remained a major focus of our work. Several small projects and a few doctoral dissertations

have dealt with the issue. This level of interest developed out of the MYCIN experience and a small group seminar series held in 1979 and 1980. Several examples of inadequate responses by MYCIN (to questions asked by users) were examined in an effort to define the reasons for suboptimal performance. One large area of problems related to MYCIN's lack of *support knowledge,* the underlying mechanistic or associational links that explain why the action portion of a rule follows logically from its premise. This limitation is particularly severe in a teaching setting where it is incorrect to assume that the system user will already know most rules in the system and merely needs to be reminded of their content. Articulation of these points was largely due to Bill Clancey's work, and they are a central element of his analysis of MYCIN's knowledge base in Chapter 29.

Other sources of MYCIN's explanation errors were its failure to deal with the context in which a question was asked (i.e., it had no sense of dialogue, so each question required full specification of the points of interest without reference to earlier exchanges) and a misinterpretation of the user's intent in asking a question. We were able to identify examples of simple questions that could mean four or five different things depending on what the user knows, the information currently available about the patient under consideration, or the content of earlier discussions. These issues are inevitably intertwined with problems of natural language understanding, and they reflect back on the second of Gorry's three concerns (language development) mentioned earlier in this chapter.

Partly as a result of work on the problem of student modeling by Bill Clancey and Bob London in the context of GUIDON, we were especially interested in how modeling the user's knowledge might be used to guide the generation of explanations. Jerry Wallis began working on this problem in 1980 and developed a prototype system that emphasized causal reasoning chains. The system associated measures of *complexity* with both rules and concepts and measures of *importance* with concepts. These reasoning chains then guided the generation of explanations in accordance with a user's level of expertise and the reasoning details that were desired. Chapter 20 describes that experimental system and defines additional research topics of ongoing interest.

Our research group continues to explore solutions to the problems of explanation in expert systems. John Kunz has developed a program called AI/MM (Kunz, 1983), which combines simple mathematical models, physiologic principles, and AI representation techniques to analyze abnormalities in fluids and electrolyte balance. The resulting system can use causal links and general laws of nature to explain physiologic observations by reasoning from first principles. The program generates English text to explain these observations.

Greg Cooper has developed a system, known as NESTOR, that critiques diagnostic hypotheses in the area of calcium metabolism. In order to critique a user's hypotheses, his system utilizes powerful explanation capabilities. Similarly, the work of Curt Langlotz, who has adapted ONCOCIN to critique a physician's therapy plan (see Chapter 32), requires

the program to explain the basis for any disagreements that occur. Langlotz has developed a technique known as hierarchical plan analysis (Langlotz and Shortliffe, 1983), which controls the comparison of two therapy plans and guides the resulting explanatory interaction. Langlotz is also pursuing a new line of investigation that we did not consider feasible during the MYCIN era: the use of graphics capabilities to facilitate explanations and to minimize the need for either typing or natural language understanding. Professional workstations and graphics languages have recently reduced the cost of high-resolution graphics systems (and the cost of programming them) enough that we expect considerably more work in this area.

Bill Clancey's NEOMYCIN research (Clancey and Letsinger, 1981), mentioned briefly in Chapter 21 and developed partially in response to his analysis of MYCIN in Chapter 29, also has provided a fertile arena for explanation research. Diane Warner Hasling has worked with Clancey to develop an explanation feature for NEOMYCIN (Hasling et al., 1983) similar to the HOW's and WHY's of MYCIN (Chapter 18). Because NEOMYCIN is largely guided by domain-independent meta-rules, however, useful explanations cannot be generated simply by translating rules into English. NEOMYCIN is raising provocative questions about how strategic knowledge should be capsulized and instantiated in the domain for explanation purposes.

Finally, we should mention the work of Randy Teach, an educational psychologist who became fascinated by the problem of explanation, in part because of the dearth of published information on the subject. Teach joined the project in 1980, discovered the issue while working on the survey of physicians' attitudes toward computer-based consultants reported in Chapter 34, and undertook a rather complex psychological experiment in an attempt to understand how physicians explain their reasoning to one another (Teach, 1984). We mention the work because it reflects the way in which the legacy of MYCIN has broadened to involve a diverse group of investigators from several disciplines. We believe that explanation continues to provide a particularly challenging set of issues for researchers from computer science, education, psychology, linguistics, philosophy, and the domains of potential application.

17.3 Current Perspective

We believe now that there are several overlapping reasons for wanting an expert system to explain its reasoning. These are

- understanding
- debugging

- education
- acceptance
- persuasion

Understanding the contents of the knowledge base and the line of reasoning is a major goal of work on explanation. Both the system builder and the user need to understand the knowledge in the system in order to maintain it and use it effectively. The system can sometimes take the initiative to inform users of its line of reasoning, such as when MYCIN prints intermediate conclusions about the type of infection or the likely identities of organisms causing a problem. More often, however, we think of a system providing explanations in response to specific requests.

The debugging rationale is important, especially because knowledge bases are built incrementally. As mentioned, this was one of Shortliffe's original motivations for displaying the rule under consideration. This line of research continues in work to provide monitoring tools within programming environments so that a system builder can watch what a system is doing while it is running. Mitch Model's Ph.D. research (Model, 1979) used MYCIN as one example for the monitoring tools he designed. His work shows the power of describing a reasoning system's activities along several different dimensions and the power of displaying those activities in different windows on a display screen.

Education is another important reason to provide insights into a knowledge base. Users who feel they learn something by interacting with an expert system are likely to use it again. As discussed in Part Eight, educating users can become as complex as providing good advice. In any case, making the knowledge base and line of reasoning understandable is a necessary step in educating users. This line of research continues in Clancey's work on NEOMYCIN (Clancey and Letsinger, 1981).

Acceptance and persuasion are closely linked. Part of making an expert system acceptable is convincing potential users and managers that its conclusions are reasonable. That is, if they understand how a system reaches conclusions on several test cases and believe that process is reasonable, they will be more likely to trust its conclusions on new cases. For the same reason, it is also important to show that the system is responsive to differences between cases.

Persuading users that a system's conclusions are correct also requires the same kind of window into the knowledge base and line of reasoning. When using a consultant program, a person is expected to understand the conclusions (and the basis for them) well enough to accept responsibility for acting on them. In medicine, for example, physicians have a moral and legal responsibility for the consequences of their actions, so they must understand why—and sometimes be persuaded that—a consultant's recommendations are appropriate.