Int. J. Man–Machine Studies (1979) **11**, 25–49

Tutoring rules for guiding a case method dialogue

WILLIAM J. CLANCEY
Computer Science Department, Stanford University, Stanford, California 94305, U.S.A.

The first version of an "intelligent computer-aided instruction" program built on MYCIN-like expert systems has been implemented. This program, named GUIDON, is a case method tutor in which the problem-solving and tutorial dialogue capabilities are distinct. The expertise to be taught is provided by a rule-based consultation program. The dialogue capabilities constitute teaching expertise for helping a student solve a case.

In this paper we describe the rule-based formalism used by MYCIN-like programs, and then argue that these programs are not sufficient in themselves as teaching tools. We have chosen to develop a mixed-initiative tutor that plays an active role in choosing knowledge to present to a student, based on his competence and interests. Furthermore, we argue that is desirable to augment the domain expertise of MYCIN-like programs with other levels of domain knowledge that help explain and organize the domain rules. Finally, we claim that it is desirable to represent teaching expertise explicitly, using a flexible framework that makes it possible to easily modify tutorial strategies and communicate them to other researchers.

The design of the GUIDON program is based on natural language studies of discourse in AI. In particular, our framework integrates domain expertise in tutorial dialogues via explicit, modular tutoring rules that are controlled by a communication model. This model is based on consideration of the student's knowledge and interests, as well as the tutor's plans for the case session. This paper discusses interesting examples of tutoring rules for guiding discussion of a topic and responding to a student's hypothesis based on the evidence he has collected.

1. Introduction

How can we make the expertise of knowledge-based programs accessible to a student? Knowledge-based programs (Davis, Buchanan & Shortliffe, 1977; Lenat, 1976; Pople, 1977; Goldstein & Roberts, 1977) achieve high performance by interpreting a specialized set of facts and domain relations in the context of particular problems. These knowledge bases are generally built by interviewing human experts to extract the knowledge they use to solve problems in their area of expertise. However, it is not clear that the organization and level of abstraction of this performance knowledge is suitable for use in a tutorial program. We are exploring this problem in the GUIDON tutorial program, using the knowledge bases of MYCIN-like expert systems.

MYCIN is a knowledge-based program that provides consultations about infectious disease diagnosis and therapy (Shortliffe, 1974). In MYCIN, domain relations and facts take the form of rules about what to do in a given circumstance. A principle feature of this formalism is the separation of the knowledge base from the interpreter for applying it. This makes the knowledge accessible for multiple uses, including application to particular problems (i.e. for "performance") and explanation of reasoning (Davis, 1976).

We have most recently used the MYCIN knowledge base as the foundation of a tutorial system, called GUIDON. The goal of this project is to study the problem of

0020-7373/79/010025+25 $02.00/0

transferring the expertise of MYCIN-like systems to a student. It is argued in this paper that MYCIN-like rule-based expert systems constitute a good basis for tutorial programs, but they are not sufficient in themselves for making knowledge accessible to a student.

In GUIDON we have augmented the performance knowledge of rules by adding two other levels: a "support level" to justify individual rules, and an "abstraction level" to organize rules into patterns. The components and tutorial uses of these levels are discussed in section 4.3. Secondly, the GUIDON system contains teaching expertise that is represented explicitly, and is independent of the domain knowledge base. This is expertise for carrying on a tutorial dialogue intended to present the domain knowledge to a student in an organized way, over a number of sessions. Section 3 describes design considerations for this tutorial dialogue, given the structure of the knowledge in MYCIN-like problem areas (described in section 2).

With the addition of other levels of domain knowledge and teaching expertise, GUIDON is designed to transfer the expertise of MYCIN-like programs in an efficient, comprehensible way. In doing this, we make contributions to several areas of research in Intelligent Computer-Aided Instruction (ICAI), including means for structuring and planning a dialogue, generating teaching material, constructing and verifying a model of what the student knows, and explaining expert reasoning.

However, we also argue that the nature of MYCIN-like expert systems makes it reasonable to experiment with various teaching strategies. The representation of teaching expertise in GUIDON is intended to provide a flexible framework for such experimentation (section 4). To illustrate the use of this framework in the first version of GUIDON, we present in this paper two sample interactions and describe the domain knowledge and teaching strategies used by the program (sections 5 and 6). The sample interactions and rule listings were generated by the implemented program.

2. Description of MYCIN-like expert systems

2.1. PROBLEM AREA AND FORMALISM

A major objective of the MYCIN system has been to provide a high performance, computer-based therapeutic tool designed to be useful in both clinical and research environments. MYCIN is a computer-based consultant that interacts with physicians in much the same way that human consultants do: it asks numerous questions about the state of the patient and provides advice about appropriate therapy. This requires development of a system that has a sound knowledge base, and that displays a high level of competence in its field.

The MYCIN knowledge base has been built over four years through interactions with physicians. This body of knowledge is represented as a collection of conditional sentences called "production rules." The production rule formalism provides a flexible and easily understood representation of facts and relations, and a simple interpreter of those facts (Davis & King, 1977). The MYCIN knowledge base currently contains approximately 450 such rules.† Each rule consists of a set of preconditions (called the

† In addition there are several hundred facts and relations stored in tables, which are referenced by the rules.

"premise") which, if true, justifies the conclusion made in the "action" part of the rule. An example is shown below.†

> IF (1) the gram stain of the organism is gram negative, 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 genus of the organism is Bacteroides.

FIG. 1. Sample MYCIN rule.

2.2. VALIDITY OF THE KNOWLEDGE BASE

Two formal evaluations of MYCIN's performance have demonstrated that MYCIN's competence in selecting antimicrobial therapy for meningitis and for bacteremia is comparable to that of the infectious disease faculty at Stanford University School of Medicine (where MYCIN was developed) (Yu *et al.*, 1978). From this we conclude that a rule-based consultant can be given the knowledge necessary for demonstrably high performance in a domain as complex and inexact as medicine. The fact that the formalism of production rules has been exploited in MYCIN to create a rich, high performance knowledge base for solving difficult, real world problems is an important starting point for demonstrating the advantages of using this representation of domain knowledge for tutoring.

2.3. DOMAIN OF APPLICATION

The production rule formalism used by MYCIN is widely applicable to tasks other than medicine, although it is by no means a "universal" language. Because the knowledge base is separate from the interpreter for applying it, it is possible to remove the medical knowledge and substitute a set of rules about a new domain.‡ The tutoring system we are developing will also work with problems and rules in another domain, assuming some parallels between the structure of the knowledge in the new domain and the structure of the existing medical knowledge. Thus, GUIDON is a multiple-domain tutorial program. The overall configuration of this system is shown in Fig. 2. (The representation of teaching expertise is discussed in section 4.) One advantage of this system is that a fixed set of teaching strategies can be tried in different domains, affording an important perspective on their generality.§

The EMYCIN system has already been used to demonstrate the applicability of the production rule formalism and interpreter to domains other than infectious disease diagnosis and therapy. For example, the SACON program‖ provides advice on structural analysis problems, such as the most appropriate materials for airplane wings under stresses of different sorts. In addition, the EMYCIN system has been used in two very

† In this paper, each precondition is called a "subgoal." If all of the subgoals in the premise can be achieved (shown to be true), then a conclusion can be made about the goal in the action.

‡ The domain-independent package, consisting of rule interpreter and explanation module is called "EMYCIN," which stands for "empty MYCIN." Development of this system is being continued by William van Melle and A. Carlisle Scott.

§ This method of integrating domain and teaching expertise can be contrasted with the design of early frame-oriented CAI systems. For example, in the tutor for infectious diseases by Feurzeig, Munter, Swets & Breen (1964), medical and teaching expertise were "compiled" together into the branching structure of the frames (dialogue/content situations). In GUIDON, domain and teaching expertise are decoupled and stated explicitly.

‖ Collaborative project with the MARC Corporation, Inc.

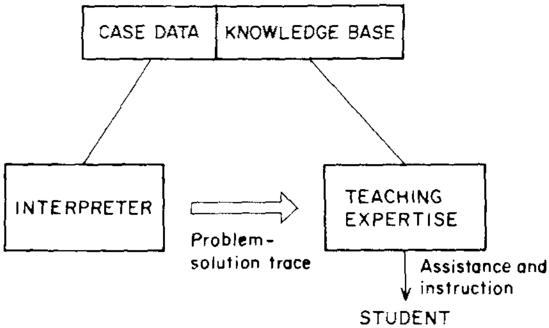

FIG. 2. Modules for a multiple-domain tutorial system.

different medical domains: interpretation of pulmonary function tests (Feigenbaum, 1977) and drug therapy recommendations for psychiatric patients (Heiser, Brooks & Ballard, 1978). It has also been used in a class exercise to diagnose and recommend fixes for problems in an automobile horn system (van Melle, 1974). In this paper, all examples and discussion will be based on the infectious disease knowledge base (the original MYCIN program).

2.4. OTHER REPRESENTATIONS FOR KNOWLEDGE BASES

Production rules have worked well in several domains (Feigenbaum, 1977).† However, other approaches for building knowledge-based systems are possible. For example, Pople successfully uses frame-like "disease hypotheses" (Pople, 1977) and Kulikowski uses a causal-associational network (Weiss, Kulikowski & Safir, 1977).

We do not argue here for or against the use of a production rule system as a foundation for tutorial programs; sounder representations of expertise may be found. But given the availability of a system with MYCIN's sophistication, there is good reason to experiment with it in a tutorial setting. It is quite possible that our use of the domain rules in a tutorial program will help us to design better formalisms for codifying expertise.

3. Development of a tutorial program based on MYCIN-like systems

In addition to the domain knowledge of the expert program, a tutorial program requires teaching expertise, such as the ability to tailor the presentation of domain knowledge to the student's competence and interests (Brown, 1977). The GUIDON program, with its teaching expertise and augmented domain knowledge, is designed to be an active, intelligent agent that helps make the knowledge of MYCIN-like programs accessible to a student.

It is possible to follow MYCIN's reasoning during a consultation by using the explanation system (one can ask WHY case data is being sought by the program and HOW goals will be (were) achieved). However, we believe that this is an inefficient process for learning the contents of the knowledge base. The MYCIN program is only a passive "teacher." It is necessary for the student to ask an exhaustive series of questions, if

† PROSPECTOR, the SRI program, was designed originally as a MYCIN-like system for consulting with geologists about mineral exploration sites.

he is to discover all of the reasoning paths considered by the program. GUIDON acts as an agent that keeps track of the knowledge that has been presented to the student in previous sessions and looks for opportunities to deepen and broaden the student's knowledge of MYCIN's expertise. Moreover, the MYCIN program contains no model of the user, so program-generated explanations are never tailored to his competence or interests. GUIDON's teaching expertise includes capabilities to measure the student's competence and to use this measure as a basis for selecting knowledge to present.

What kind of dialogue might be suitable for teaching the knowledge of MYCIN-like consultation systems? What strategies for teaching will be useful? Will these strategies be independent of the knowledge base content? How will they be represented? What additions to the performance knowledge of MYCIN-like systems might be useful in a tutorial program? These are some of the basic questions involved in converting a rule-based expert program into a tutorial program.

As the first step in approaching these questions, the following subsections discuss some of the basic ways in which MYCIN's domain and formalism have influenced design considerations for GUIDON. Section 3.1 describes the nature of the dialogue we have chosen for tutorial sessions. Section 3.2 discusses the nature of MYCIN performance knowledge and argues for including additional domain knowledge in the tutorial program. Sections 3.3 and 3.4 argue that the uncertainty of MYCIN's knowledge and the size of the knowledge base make it desirable to have a framework for experimenting with teaching strategies. This framework, which describes GUIDON as a discourse program and will provide the basis for future development of the system, is presented in section 4.

3.1. A GOAL-DIRECTED CASE DIALOGUE

In a GUIDON tutorial session, a student plays the role of a physician consultant. A sick patient (the "case") is described to him in general terms: his age, sex, race and lab reports about cultures taken at the site of the infection are provided. The student is expected to ask for other information that he thinks might be relevant to this case: e.g. did the patient become infected while hospitalized? Did he ever live in the San Joaquin Valley? GUIDON compares the student's questions to those asked by MYCIN and critiques him on this basis. When the student draws hypotheses from the evidence he has collected, GUIDON compares these conclusions to those that MYCIN reached, given the same information about the patient. We refer to this dialogue between the student and GUIDON as a "case dialogue." Because GUIDON attempts to transfer expertise to students exclusively through case dialogues, we call it a "case method tutor."

It is assumed that the student wishes to learn to solve the problems which MYCIN can solve. GUIDON's purpose is to broaden the student's knowledge of the evidence to consider in a particular problem by pointing out inappropriate lines of reasoning and suggesting approaches the student did not consider. An important assumption is that the student has a suitable background for solving the case; he knows the vocabulary and the general form of the task (using case data to reach a diagnosis). GUIDON can help him judge the relative importance of the evidence in specific cases. The criterion for having learned MYCIN's problem-solving methods is therefore straightforward: when presented with novel, difficult cases, does the student seek relevant data and draw appropriate conclusions?

Helping the student solve the case is greatly aided by placing constraints on the case dialogue. A "goal-directed" dialogue is a discussion of the rules applied to achieve specific goals. In general, the topics of this dialogue are precisely those "goals" that are concluded by MYCIN rules.† During the dialogue, only one goal at a time is considered; data that cannot be used in rules to achieve this goal are "irrelevant." This is a strong constraint on the student's process of asking questions and making hypotheses, A goal-directed dialogue helps the tutor to follow the student as he solves the problem, increasing the chance that timely assistance can be provided.‡

Our design of GUIDON has also been influenced by consideration of the sophistication of the students we expect to use it. We are designing the program for well-motivated students who are capable of a serious, mixed-initiative dialogue. Various features (not all described in this paper) make the program flexible, so that students can use their judgment to control the depth and detail of the discussion. These features include the capability to request:

(1) a list of descriptions of all data relevant to a particular goal;
(2) a subgoal tree for a goal;
(3) a quiz or hint relevant to the current goal;
(4) a concise summary of all evidence already discussed for a goal (data and rules that were mentioned in the dialogue);
(5) discussion of a goal (of the student's choice);
(6) conclusion of a discussion, with GUIDON finishing the collection of evidence for the goal, and indicating conclusions that the student might have drawn.

3.2. SINGLE FORM OF EXPERTISE

The problem of multiple forms of expertise has been important in ICAI research. For example, when mechanistic reasoning is involved, qualitative and quantitative forms of expertise may be useful to solve the problem (Brown, Rubenstein & Burton, 1978). de Kleer has found that strategies for debugging an electronic circuit are "radically different" depending on whether one does local mathematical analysis (using Kirchhoff's laws) or uses a higher level, functional analysis of components (Brown *et al.*, 1975). One might argue that a tutor for this domain should be ready to recognize and generate arguments on both of these levels.§

For all practical purposes, GUIDON does not need to be concerned about multiple forms of expertise. This is primarily because reasoning in infectious disease problem-solving is based on judgments about empirical information, rather than arguments based on causal mechanisms (Weiss *et al.*, 1977). MYCIN's judgments are "cookbook" responses that address the data directly, as opposed to attempting to explain it in terms of physiological mechanisms. Moreover, the expertise to solve a MYCIN case on this level of abstraction constitutes a "closed" world (Carbonell & Collins, 1977): all of the

† A typical sequence of (nested) goals is: to reach a diagnosis, to determine which organisms might be causing the infection, to determine the type of infection, to determine if the infection has been partially treated, etc.

‡ Sleeman (1977) uses a similar approach for allowing a student to explore algorithms.

§ See Carr & Goldstein (1977) for related discussion.

objects, attributes and values that are relevant to solution of a case are determined by a MYCIN consultation that is performed before a tutorial session begins†

It is possible that multiple forms of expertise might be applied to solve some of the problems in domains of other MYCIN-like expert systems, e.g. in solving structural analysis problems. Using GUIDON in these domains may suggest ways to use the production rule formalism to encode multiple forms of expertise.

Even though MYCIN's domain makes it possible for cases to be solved without recourse to the level of physiological mechanisms, a student may find it useful to know this "support" knowledge that lies behind the rules. Section 4.3 describes the domain knowledge we have found it useful to add to MYCIN's performance knowledge in GUIDON.

3.3. WEAK MODEL OF INQUIRY

Even though the MYCIN world can be considered to be closed, there is no strong model for ordering the collection of evidence.‡ Medical problem solving is still an art. While there are some conventions that ensure that all routine data is collected, physicians have not agreed upon a basis for numerically optimizing the decision of what to do next.§ For example, when offering assistance, should the tutor suggest the domain rule that most confirms the evidence that has already been collected, or a rule that contradicts this evidence?‖ It will be useful to experiment with various strategies for guiding the student's collection of case evidence.

3.4. LARGE NUMBER OF RULES

For every case GUIDON discusses with a student, MYCIN provides an "AND/OR" tree of goals (the "OR" nodes) and rules (the "AND" nodes) that were pursued during the corresponding consultation. This tree constitutes a trace of the application of the knowledge base to the given case.¶ Many of the 450 rules are not tried because they conclude about goals that don't need to be pursued to solve the case. Hundreds of others fail to apply because one or more preconditions are found to be false. Finally,

† There is always the possibility that a student may present an exotic case to the GUIDON that is beyond its expertise. While MYCIN has been designed to detect simple instances of this (i.e. evidence of an infection other than bacteremia or meningitis), we have decided to restrict GUIDON tutorials to the physician-approved cases in the library (currently over 100 cases).

‡ In Goldstein's WUMPUS program (Carr & Goldstein, 1977), for example, it is possible to rank each legal move (analogous to seeking case data in MYCIN) and so rate the student according to "rejected inferior moves" and "missed superior moves." The same analysis is possible in Burton and Brown's WEST program (Brown *et al.*, 1975).

§ See, for example, Sprosty (1963).

‖ MYCIN's rules are not based on Bayesian probabilities, so it is not possible to use optimization techniques like those developed by Hartley (Hartley, Sleeman & Woods, 1972). Arguments against using Bayes Law in expert systems can be found in Shortliffe (1974).

¶ Before a tutorial session, GUIDON scans each rule used by MYCIN and compiles a list of all subgoals that needed to be achieved before the premise of the rule could be evaluated. In the case of a rule that failed to apply, GUIDON determines all preconditions of the premise that are false. By doing this, GUIDON's knowledge of the case is independent of the order that questions were asked and rules were applied by MYCIN, so topics can be easily changed and the depth of discussion controlled flexibly by both GUIDON and the student. The procedure for constructing the trace is quite complicated and not all of the problems have been solved. Details will appear in later publications. This process of automatically generating a solution trace for any case can be contrasted with SOPHIE's single, fixed simulated circuit (Brown *et al.*, 1976).

typically 20% of the rules make conclusions that contribute varying degrees of belief about the goals pursued.

Thus, MYCIN interpreter provides the tutorial program with a lot of information about the case solution (see Fig. 2). It is not clear how to present this to a student. What should the tutor do when the student pursues a goal that MYCIN did not? (Interrupt him? Wait until he realizes that the goal contributes no useful information?) Which dead-end search paths pursued by MYCIN should the tutor expect the student to consider? For many goals there are too many rules to discuss with the student; how is the tutor to decide which to present and which to omit? What techniques can be used to produce coherent plans for guiding the discussion through lines of reasoning used by the program? It would be useful to have a framework that gave us the freedom to guide the dialogue in different ways. The rest of this paper shows how GUIDON has been given this flexibility by viewing it as a discourse program.

4. A framework for a case method tutorial program

One purpose of this tutorial project is to provide a framework for testing teaching methods. Therefore, we have chosen an implementation that makes it possible to vary the strategies that the tutor uses for guiding the dialogue. Using methods similar to those used in knowledge-based programs, we have formalized the tutorial program in rules and procedures that codify expertise for carrying on a case dialogue.

This section is relatively abstract. The reader may find it useful to consider the sample dialogues in Figs 7 and 8 before proceeding. The first subsection below lists forms of discourse knowledge that will be useful for a case method dialogue. Following subsections describe GUIDON's representation of this knowledge. Examples and details are provided in sections 5 and 6.

4.1. DISCOURSE KNOWLEDGE

Our implementation of GUIDON's dialogue capabilities makes use of knowledge obtained from studies of discourse in AI (Bobrow *et al.*, 1977; Bruce, 1975; Deutsch, 1974; Winograd, 1977). To quote Bruce (emphasis added):

> [It is] . . . useful to have a model of how social interactions typically fit together, and thus a model of discourse structure. Such a model can be viewed as a heuristic which suggests likely *action sequences*. . . There are places in a discourse where questions make sense, others where explanations are expected. [These paradigms]. . . facilitate generation and subsequent understanding (Bruce, 1975).

Based on Winograd's analysis of discourse (Winograd, 1977), it appears desirable for a case method tutor to have the following forms of knowledge for carrying on a dialogue.

(1) Knowledge about dialogue patterns. Faught (1977) mentions two types of patterns: interpretation patterns (to understand a speaker) and action patterns (to generate utterances). GUIDON uses action patterns represented as "discourse procedures" for directing and focussing the case dialogue. These are the "action sequences" mentioned by Bruce. They are invoked by tutoring rules, discussed in section 4.2.†

† Because of the constraints a goal-directed dialogue imposes upon the student, we have not found it necessary to use interpretation patterns at this time. They might be useful to follow the student's reasoning in a non-goal-directed dialogue.

(2) Forms of domain knowledge for carrying on a specific dialogue, Section 4.3 surveys the augmented domain knowledge available to GUIDON.
(3) Knowledge of the communication situation. This includes the tutorial program's understanding of the student's intentions and knowledge, as well as the tutor's intentions for carrying on the dialogue. These components are represented in GUIDON by an "overlay student model" (in which the student's knowledge is viewed as a subset of the expert program's) (section 4.4.1), a "case syllabus" (a lesson plan of topics to be sure are discussed, created by the tutor for each case) (section 4.4.2), and a "focus record" (to keep track of factors in which the student has shown interest recently) (section 4.4.3). Knowledge of the communication situation controls the use of dialogue patterns.

The following subsections give details about each of these forms of knowledge.

4.2. DISCOURSE PROCEDURES AND TUTORING RULES

The sequences of actions in discourse procedures serve as an ordered list of options—types of remarks for the program to consider making. For example, the procedure for discussing a domain rule (hereafter, d-rule) includes a step that indicates to "consider mentioning d-rules related to the one just discussed." Thus, a discourse procedure step specifies in a schematic form WHEN a type of remark might be appropriate. WHETHER to take the option (e.g. is there an "interesting" d-rule to mention?) and WHAT to say exactly (the discourse pattern for mentioning the d-rule) will be dynamically determined by tutoring rules (hereafter, t-rules) whose preconditions refer to the student model, case syllabus, and focus record (hereafter, referred to jointly as the communication model).

T-rules are generally invoked as a packet to achieve some tutorial goal.† T-rule packets are of two types:

(1) T-rules for accumulating belief—updating the communication model and determining how "interesting" a topic is are two examples.‡ Generally, a packet of t-rules of this type is applied exhaustively.
(2) T-rules for selecting a discourse procedure to follow. Generally, a packet of this type stops trying t-rules when the first one succeeds. The form of t-rules of this type is shown in Fig. 3. Knowledge referenced in the premise part of a t-rule of this type is described in subsequent sections. The action part of these t-rules consists of stylized code, just like the steps of a discourse procedure.§ A step may invoke:
    (a) a packet of t-rules, e.g., to select a question format for presenting a given d-rule;
    (b) a discourse procedure, e.g., to sequentially discuss each precondition of a d-rule;

† Packets are implemented as stylized INTERLSIP procedures. This should be contrasted with the interpreter used by the expert program that invokes d-rules directly, indexing them according to the goal that needs to be determined.

‡ GUIDON uses "certainty factors" for representing the program's belief in something. Their value ranges between −1 and 1, with negative values signifying disbelief. See Shortliffe & Buchanan (1975) for discussion of their implementation and significance.

§ Discourse procedure steps also contain control information (e.g. for iteration) that is not important to this discussion.

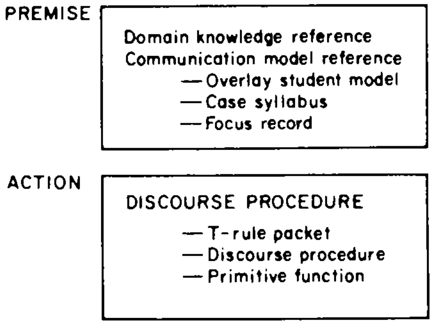

FIG. 3. Form of a tutorial rule.

(c) a primitive function, e.g. to accept a question from the student, perform bookkeeping, etc.

Below is an outline of the t-rules currently being implemented in GUIDON. Except where noted, examples of these t-rules are presented in discussions of the sample tutorial dialogues in this paper.

I. *T-rules for selecting discourse patterns*
   A. Guiding discussion of a d-rule.
   B. Responding to a student hypothesis.
   C. Choosing question formats.

II. *T-rules for choosing domain knowledge*
   A. Providing orientation for pursuing new goals (not demonstrated in this paper).
   B. Measuring interestingness of d-rules.

III. *T-rules for maintaining the communication model*
   A. Updating the overlay model when d-rules fire.
   B. Updating the overlay model during hypothesis evaluation.
   C. Creating a case syllabus (not implemented).

All of the t-rules in this paper are translated by a program directly from the INTER-LISP source code, using an extension of the technique used for translating MYCIN's rules. This accounts for some of the stilted prose in the examples that follow.

4.3. AUGMENTED REPRESENTATION OF DOMAIN KNOWLEDGE

The representation of domain knowledge available to GUIDON can be organized in three tiers, shown by Fig. 4.

Subsequent sections briefly describe the components of each tier. Section 4.3.4 discusses how meta-level knowledge that describes the representation of MYCIN-like rules has been used in GUIDON for implementing a variety of tutorial strategies.

4.3.1. Performance tier

The performance knowledge consists of all the rules and tables used by the expert program to make goal-directed conclusions about the initial case data. The output of

I. META-LEVEL ABSTRACTIONS: rule models
rule schemata

II. PERFORMANCE: rules
lists and tables

III. SUPPORT: definitions
mechanism descriptions
justifications
literature references

FIG. 4. Domain knowledge organization into three tiers.

the consultation, an extensive AND/OR tree of traces showing which rules were applied, their conclusions, and the case data required to apply them, is passed to the tutor. GUIDON fills in this tree by determining which subgoals appear in the rules. In Fig. 5, "COVERFOR" signifies the goal to determine which organisms should be "covered" by a therapy recommendation; d-rule 578 concludes about this goal; "BURNED" is a subgoal of this rule.

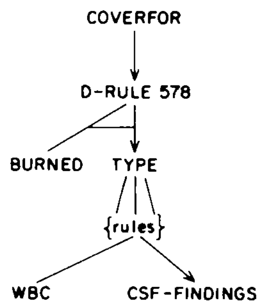

FIG. 5. The portion of the AND/OR tree of goals and rules created by the expert program which is relevant to the dialogue shown in Fig. 7.

Tutorial rules make frequent reference to this data structure in order to guide the dialogue. For example, the response to the request for help shown in Fig. 7 is based first of all on the rules that were used by the expert program for the current goal. Similarly, the t-rules for supplying case data requested by the student check to see if the expert program asked for the same information during the corresponding consultation (e.g. the "white blood count" in the sample dialogue of Fig. 7).†

D-Rule578 and its associated documentation is illustrated in Fig. 6.

4.3.2. Support tier

The support tier of the knowledge base consists of annotations to the rules and the factors used by them.‡ For example, there are canned-text descriptions of every

† Other possibilities include: the question is not relevant to the current goal; it can be deduced by definition from other known data; or a d-rule indicates that it is not relevant to this case.

‡ Rule justifications, author and edit date were first proposed by Davis (1976) as knowledge base maintenance records.

⟨abstraction level⟩

RULE-SCHEMA: MENINGITIS . COVERFOR . CLINICAL
RULE-MODEL: COVERFOR-IS-MODEL
KEY-FACTOR: BURNED
DUAL: D-RULE577

⟨performance level⟩

D-RULE578

If: (1) The infection which requires therapy is meningitis, and
(2) Organisms were not seen on the stain of the culture, and
(3) The type of the infection is bacterial, and
(4) The patient has been seriously burned
Then: There is suggestive evidence (0·5) that pseudomonas-aeruginosa is one of the organisms (other than those seen on cultures or smears) which might be causing the infection
UPDATES: COVERFOR
USES: (TREATINF ORGSEEN TYPE BURNED)

⟨support level⟩

MECHANISM-FRAME: BODY-INFRACTION . WOUNDS
JUSTIFICATION: "For a very brief period of time after a severe burn the surface of the wound is sterile. Shortly thereafter, the area becomes colonized by a mixed flora in which Gram pos organisms predominate. By the 3rd post burn day this bacterial population becomes dominated by Gram neg organisms. By the 5th day these organisms have invaded tissue well beneath the surface of the burn. The organisms most commonly isolated from burn patients are *Pseudomonas*, *Klebsiella-Enterobacter*, *Staph*. etc. Infection with *Pseudomonas*. is frequently fatal."
LITERATURE: Macmillan BG: Ecology of Bacteria Colonizing the Burned Patient Given Topical and System Gentamicin Therapy: a five-year study, J Infect Dis 124:278–286, 1971.
AUTHOR: Dr. Victor Yu
LAST-CHANGE: 8 September, 1976

FIG. 6. Domain Rule 578 and its associated documentation. All information is provided by a domain expert, except for the "key-factor" which is computed by the tutor from the rule schema and contents of the particular rule. See section 4.3.3.

laboratory test in the MYCIN domain, including, for instance, remarks about how the test should be performed. Mechanism descriptions provided by the domain expert are used to provide some explanation of a rule beyond the canned text of the justification. For the infectious disease domain of MYCIN, they indicate how a given factor leads to a particular infection with particular organisms by stating the origin of the organism and the favourable conditions for its growth at the site of the infection. Thus, the frame associated with the factor "a seriously burned patient" shows that the organisms originate in the air and grow in the exposed tissue of a burn, resulting in a frequently fatal infection.

4.3.3. Abstraction tier

The abstraction tier of the knowledge base represents patterns in the performance knowledge. For example, a rule schema is a description of a kind of rule: a pattern of preconditions that appears in the premise, the goal concluded about, and the context of its application. The schema and a canned-text annotation of its significance are formalized in the MYCIN knowledge base by a physician expert. This schema is used by the tutor to "subtract off" the rule preconditions common to all rules of the type, leaving behind the factors that are specific to this particular rule, i.e. the "key factors" of this rule. Thus, the key factor of d-rule 578 (see Fig. 6), the fact that the patient has been seriously burned, was determined by removing the "contextual" information of the name of the infection, whether organisms were seen, and the type of the infection. Examples of the use of key factors occur throughout the hypothesis evaluation example (Fig. 8), particularly in lines 4–9.

Rule models (Davis, 1976) are program-generated patterns that represent the typical clusters of factors in the expert's rules. Unlike rule schema, rule models do not necessarily correspond to domain concepts, though they do represent factors that tend to appear together in domain arguments (rules). An example from the MYCIN data base shows that the gram stain of an organism and its morphology tend to appear together in rules for determining the identity of an organism. Because rule models capture the factors that most commonly appear in rules for pursuing a goal, we are experimenting with their use as a form of "orientation" for naïve students. Details will be provided in later publications.

4.3.4. Use of meta-knowledge in tutorial rules

Meta-knowledge of the representation and application of d-rules plays an important role in t-rules. For example, in the first dialogue excerpt (Fig. 7) GUIDON uses function templates† to "read" d-rule 578 and discovers that the type of the infection is a subgoal that needs to be completed before the d-rule can be applied. This capability to examine the domain knowledge and reason about its use enables GUIDON to make multiple use of any given production rule during the tutorial session. Here are some uses we have implemented.

(1) Examine the rule (if it was tried in the consultation) and determine the subgoals that needed to be achieved before it could be applied; if the rule failed to apply, determine all possible ways this could be determined (perhaps more than one precondition is false).
(2) Examine the state of application of the rule during a tutorial interaction (what more needs to be done before it can be applied?) and choose an appropriate method of presentation.

† A function's template "indicates the order and generic type of the arguments in a typical call of that function" (Davis & Buchanan, 1977).

(3) Generate different questions for the student.
(4) Use the rule (and variations of it) to understand a student's hypothesis.
(5) Summarize arguments using the rule by extracting the "key point" it addresses.

The ability to use domain knowledge in multiple ways is an important feature of a "generative" tutor like GUIDON.† Flexible use of knowledge permits us to write a variety of tutoring rules that select and present teaching material in multiple ways. This is important because we want to use the MYCIN/GUIDON system for experimenting with teaching strategies.

4.4. COMPONENTS OF THE COMMUNICATION MODEL

The components of the commuication model are (1) an overlay student model, (2) a case syllabus and (3) a focus record.

4.4.1. The overlay student model

The d-rules that were fired during the consultation associated with this case are run in a forward direction as the student is given case data.‡ In this way, GUIDON knows at every moment what the expert program would conclude based on the evidence available to the student. We make use of knowledge about the history and competence of the student to form hypotheses about which of the expert's conclusions are probably known to the student. This has been termed an "overlay" model of the student by Goldstein, because the student's knowledge is modelled in terms of a subset and simple variations of the expert rule base (Goldstein, 1977). Our work was originally motivated by the structural model used in Burton and Brown's WEST system (Brown *et al.*, 1975).

Special t-rules for updating the overlay models are invoked whenever the expert program successfully applies a d-rule. These t-rules must decide whether the student has reached the same conclusion. This decision is based upon:

(1) the inherent complexity of the d-rule (e.g., some rules are trivial definitions, others have involved iterations);
(2) whether the tutor believes that the student knows how to achieve the subgoals that appear in the d-rule (factors that require the application of rules);
(3) the background of the student (e.g. year of medical school, intern, etc.);
(4) evidence gathered in previous interactions with the student.

These considerations are analogous to those used by Carr & Goldstein (1977) for the WUMPUS tutor.

4.4.2. The case syllabus

Before a human tutor discusses a case with a student, he has an idea of what he wants to discuss, given the constraints of time and the student's interests and capabilities. Similarly, in later versions of GUIDON a case syllabus will be generated before each

† Generative CAI programs select and transform domain knowledge in order to generate individualized teaching material. See Koffman & Blount (1973) for discussion.

‡ This is one application of the problem solution trace. The structure of this trace permits the program to repetitively reconsider d-rules (indexing them by the case data referenced in the premise part), without the high cost of reinterpreting premises from scratch.

case session.† We would like the syllabus to give GUIDON a global sense of the value of discussing particular topics, especially as depth of emphasis will impact on the student's understanding of the problem's solution. The syllabus of the type we are proposing provides consistency and goal-directedness to the tutor's presentations.

The syllabus will be derived from the following:

(1) The student model—where does the student need instruction?
(2) Professed student interests (perhaps the case was chosen because of features the student wants to know more about).
(3) Intrinsic importance of topics: what part does this information play in understanding the solution of the problem?
(4) Extrinsic importance of topics: given the universe of cases, how interesting is this topic? (A datum that is rarely available is probably worth mentioning when it is known, no matter how insignificant the evidence it contributes.)

We believe that these considerations will also be useful for implementing automatic selection of cases from the consultation library.

4.4.3. The focus record

The purpose of the focus record is to maintain continuity during the dialogue. It consists of a set of global variables that are set when the student asks about particular goals and values for goals. T-rules reference these variables when selecting d-rules to mention or when motivating a change in the goal being discussed. An example is provided in section 5.1.

5. T-rules for guiding discussion of a goal

In this section we consider an excerpt from a dialogue and some of the discourse procedures and tutoring rules involved. Suppose that a first-year medical student has just read about treatment for burned patients suspected to have a meningitis infection. His microbiology text mentioned several organisms, but it wasn't clear to him how other factors such as the age and degree of sickness of the patient might affect diagnosis of an actual case. GUIDON is available to him, so he decides to ask the program to select a relevant case from the MYCIN library for a tutorial session.

The program begins by invoking the discourse procedure CASE-DISCUSSION. One of the first steps is to choose a case. At this point the student described the case he wanted using keywords ("burned meningitis patient").‡ GUIDON selected the case and set the scene for the student. There is one pending culture from the CSF (cerebral spinal fluid, where meningitis infections occur). In the excerpt below (Fig. 7), the dialogue has reached the subtask of determining the organisms which might be causing

† Goldstein's "syllabus" and BIP's "Curriculum Information Network" are fixed networks that relate skills in terms of their complexities and dependencies. The syllabus discussed here is a program-generated plan for guiding discussion of a particular problem with a particular student.

‡ Student input to the GUIDON program is in the form of menu options and simple English phrases that are parsed using keyword analysis and pattern matching routines developed for MYCIN's question-answering module (Scott *et al.*, 1977; Shortliffe, 1974).

the infection. (Annotations appear in curly brackets { }; student input follows double asterisks; line numbers are referred to in subsequent discussion of this example.

.
.
.

You should now proceed to ask questions that will enable you to make a hypothesis about the organisms (other than those seen on cultures or smears) which might be causing the infection.

{Initiative is given to the student; he may ask for more data; make a hypothesis; or use a special option.}

** HOW IS THE FACT THAT THE PATIENT IS BURNED RELEVANT?

{One option is to ask how a piece of data is relevant to the current goal.}

The fact that patient-326 is seriously burned will be used to conclude that *Pseudomonas aeruginosa* is one of the organisms which might be present (in addition to those seen on cultures or smears) [Rule 578].

** HELP

{The "help" option is a request for assistance: the student asks "where do I go from here?"}

Try to determine the type of the infection: bacterial, fungal, viral, or TB.

{To provide help, GUIDON chose: (1) a rule to discuss (D-rule 578 was chosen because the dialogue just focused on it); and (2) a method of presentation—"tell the student to finish incomplete subgoals mentioned in the rule." Thus, a new topic is set up, "the type of the infection." Initiative is returned to the student . . .}

** WHAT IS THE PATIENT'S WBC?

The white count from the patient's peripheral CBC is 1·9 thousand.

{The tutor observes that this question is relevant to the new topic and provides the requested data. The dialogue continues in this goal-directed manner . . .}

.
.
.

FIG. 7. Sample interaction: gathering data.

When the student requested help (line 17), the program had been following the pattern for discussing a goal. The request for help led to the invocation of tutoring rules. The teaching strategy represented by these t-rules is to provide help for a goal by suggesting a d-rule to the student. The discourse procedure that provides help in this context first invokes a packet of t-rules that will choose a d-rule to mention to the student. The second step is to invoke a packet of t-rules that will choose a presentation method.

5.1. CHOOSING A D-RULE TO MENTION IN ORDER TO PROVIDE HELP FOR A GOAL

D-rule 578 (see Fig. 6) was chosen because it became the focus of the discussion when the student asked about the relevance of the "burned" factor. That is, when the student asked the question in line 8, a variable was set to indicate that the most recent factor referred to for this goal was "burned" (the "focus topic"). Then when the packet of t-rules for choosing a d-rule to present was invoked, the following t-rule succeeded:†

T-RULE26.03

If: The recent context of the dialogue mentioned either a "deeper subgoal" or a factor relevant to the current goal
Then: Define the focus rule to be the d-rule that mentions this focus topic

This example illustrates how the communication model guides the session by controlling t-rules.

Often there is no obvious d-rule to suggest to the student. It is then useful for the tutor to have some measure of the "interestingness" of a d-rule at this time in the discussion. The t-rules presented below are applied to a set of d-rule candidates, ranking them by how strongly the tutor believes that they are interesting.

5.1.1. Change in belief is interesting

One measure of interest is the contribution the d-rule would make to what is currently known about the goal being discussed. If the d-rule contributes evidence that raises the certainty of the determined value of the goal to more than 0·2, we say that the value of the goal is now significant.‡ This contribution of evidence is especially interesting because it depends on what evidence has already been considered.

Like all t-rules, this determination is a heuristic which will benefit from experimentation. In t-rule 25.01 we have attempted to capture the intuitive notion that, in general, change in belief is interesting: the more drastic the change, the more interesting the effect. The numbers in the conclusion of t-rule 25.01 are certainly factors that indicate our relief in this interestingness.

T-RULE25.01

If: The effect of applying the d-rule on the current value of the goal has been determined
Then: The "value interest" of this d-rule depends on the effect of applying the d-rule as follows:
a. if the value contributed is still insignificant then 0·05
b. if a new insignificant value is contributed then 0·05
c. if a new significant value is contributed then 0·50
d. if a significant value is confirmed then 0·70
e. if a new strongly significant value is contributed then 0·75
f. if an insignificant value becomes significant then 0·80
g. if an old value is now insignificant then 0·85
h. if belief in an old value is strongly contradicted then 0·90

† T-rule numbers are of the form: ⟨procedure number that invokes the rule⟩ . ⟨index of the rule⟩. Thus, t-rule 26.03 is the third rule in discourse procedure number 26.

‡ For example, if the goal is the "organism causing the infection" and the certainty associated with the value "*Pseudomonas*" is 0·3, then this value is significant.

5.1.2. Use of special facts or relations is interesting

In contrast to t-rule 25.01, the measure of interest in t-rule 25.06 below is static. We would like to make sure that the student knows the information in tables used by the expert program, so we give a d-rule that references a table special consideration.

T-RULE25.06

If: The d-rule mentions a static table in its premise
Then: Define the "content interest" to be 0·50

5.2. GUIDING DISCUSSION OF A D-RULE

Returning to our example, after selecting d-rule 578, the tutor needed to select a method for presenting it. The following t-rule was successfully applied:†

T-RULE2.04

If: (1) The number of factors appearing in the d-rule which need to be asked by the student is zero, and
(2) The number of subgoals remaining to be determined before the d-rule can be applied is equal to 1
Then: Substep i. Say: subgoal-suggestion
Substep ii. Discuss the goal with the student in a goal-directed mode [Proc001]
Substep iii. Wrap up the discussion of the rule being considered [Proc017]

The premise of this t-rule indicates that all preconditions of the d-rules can be evaluated, save one, and this d-rule precondition requires that other d-rules be considered. The action part of this t-rule is a sequence of actions to be followed, i.e. a discourse pattern. In particular, substep "i." resulted in the program printing "try to determine the type of the infection . . ." (line 22). The discourse procedure invoked by substep "ii." will govern discussion of the type of the infection (in simple terms, a new context is set up for interpreting student questions and use of options). After the type of the infection is discussed (relevant data is collected and hypotheses drawn), the tutor will direct the dialogue to a discussion of the conclusion to be drawn from d-rule 578.

Other methods for "suggesting a d-rule" are possible and are selected by other t-rules in the packet that contained t-rule 2.04. For example, the program could simply tell the student the conclusion of the d-rule (if the d-rule can be evaluated based on data currently available to the student), or quiz the student about the d-rule, or sequentially discuss each precondition of the d-rule, and so on.

6. T-rules for responding to a student's hypothesis

A student's hypothesis provides the tutor with considerable evidence about his knowledge of the d-rules. It is an excellent opportunity for the tutor to update the model of the student's understanding and use that model to make instructive remarks. The chief strategy behind the tutorial remarks illustrated below is to state evidence directly or to use simple true-false questions when the overlay model shows that student is probably ignorant of the rules to apply (lines 11–17), otherwise use fill-in and multiple choice questions that are intended to "entrap" the student (lines 19–45).‡

† "Say: ⟨label⟩" designates something the program will "say" to the student. The label is useful for debugging, because every "print statement" is uniquely labelled.

‡ Please note that lines 4–6 and 41–43 have been paraphrased to illustrate work in progress.

⋮

{After presentation of relevant case data . . .}

** I THINK THAT THE ORGANISMS PRESENT ARE DIPLOCOCCUS-PNEUMONIAE, PSEUDOMONAS-AERUGINOSA, AND NEISSERIA-MENINGITIDIS.

Good. The age of the patient gives evidence for concluding that Diplococcus-Pneumoniae might be present (Rule 507). The fact that the patient has been seriously burned is good reason to suspect *Pseudomonas aeruginosa* (Rule 578). Both conclusions apply because the infection which requires therapy is meningitis, no organisms were seem on a gram stain of the culture, and the type of the infection is bacterial.

The age of the patient also leads one to suspect *Neisseria meningitidis* (Rule 507). Do the fact that the infection was acquired while the patient was hospitalized and information about the diseases to which the patient has been exposed support this conclusion? ** NO

Right, this would be good reason to rule out *Neisseria meningitidis* (Rule 561).

What facts about this case tell you that the organisms which might be present are: *E. coli* and *Klebsiella pneumoniae*?
** HELP
{The student was expected to describe each factor with a phrase, one per line. Help is provided by listing all relevant factors for the current goal.}
Choose from:
1. The patient does not have a TB risk factor and the patient's x-ray is not abnormal
2. Otitis-media and Epigolottitis are not among the diagnoses which are consistent with the patient's clinical history
3. The white count from the patient's peripheral CBC is less than 2·5 thousand
4. The infection was acquired while the patient was hospitalized
5. The patient has been seriously burned
6. None of these

** 1 3 4 {This question was a form of "entrapment." Notice that the choices all appear in Fig. 9.}

You're right about the WBC (Rule 557) and the evidence gained from the fact that the infection was acquired while the patient was hospitalized (Rule 545). However, the lack of a TB risk factor and a normal x-ray is weak evidence that Mycobacterium-TB is not one of the organisms which might be present (Rule 160).

FIG. 8. Sample interaction: hypothesis evaluation.

"Entrapment," as used here, involves forcing the student to make a choice that will reveal some aspect of his understanding.† In this example, all choices listed (lines

† Many of GUIDON's question-asking strategies resemble the "Socratic" strategies formalized by Collins (1976), probably because our production rule representation of domain knowledge makes it convenient to think in terms of "relevant factors" for a determining the "value of a goal" (terms we share with Collins). However, the relation between factor and goal in MYCIN is not necessarily causal as it is in the network representation used by Collins.

26–33) actually appear in rules applied by MYCIN (see Fig. 9). When the student wrongly chose number 1 ("TB risk factor and abnormal x-ray"), GUIDON indicated how that evidence actually was used by MYCIN.

6.1. UPDATING THE OVERLAY STUDENT MODEL AFTER A STUDENT HYPOTHESIS

Figure 9 illustrates how the overlay model is updated for the hypothesis in Fig. 8. T-rules are invoked to determine how strongly the tutor believes that the student has taken each of the relevant d-rules into account. That is, a packet of t-rules (packet number 6 here) is tried in the context of each d-rule. Those t-rules that succeed will modify the cumulative belief that the given d-rule was considered by the student. T-rule 6.05 succeeded when applied to d-rules 545 and 557. While the student mentioned a value that they conclude (pseudomonas) (clause 1 of the t-rule), he missed others (clause 3). Moreover, he did not mention values that can ONLY be concluded by these d-rules (clause 2), so the overall evidence that these d-rules were considered is weak (−0·70).†

T-RULE6.05

If: (1) The hypothesis does include values that can be concluded by this d-rule, as well as others, and
(2) The hypothesis does not include values that can only be concluded by this d-rule, and
(3) Values concluded by the d-rule are missing in the hypothesis
Then: Define the belief that the d-rule was considered to be −0·70.

After each of the d-rules applied by MYCIN is considered independently, second pass is made to look for patterns. Two judgmental tutorial rules from this second rule packet are shown below. T-rule 7.01 applied to d-rule 578: of the d-rules that conclude pseudomonas, this is the only one that is believed to have been considered, thus increasing our belief that d-rule 578 was used by the student. T-rule 7.05 applies to d-rules 545 and 561: the factor NOSOCOMIAL appears only in their premises, and they are not believed to have been considered. This is evidence that NOSOCOMIAL was not considered by the student, increasing our belief that each of the d-rules that mention it were not considered.

T-RULE7.01

If: You believe that this domain rule was considered, it concludes a value present in the student's hypothesis, and no other rule that mentions this value is believed to have been considered
Then: Modify the cumulative belief that this rule was considered by 0·40

T-RULE7.05

If: This domain rule contains a factor that appears in several rules, none of which are believed to have been considered to make the hypothesis
Then: Modify the cumulative belief that this rule was considered by −0·30

Future improvements to this overlay model will make it possible to recognize student behavior that can be explained by simple variations of the expert's d-rules.

† The certainty factor of −0·70 was chosen by the author. Experience with MYCIN shows that the precise value is not important, but the scale from −1 to 1 should be used consistently.

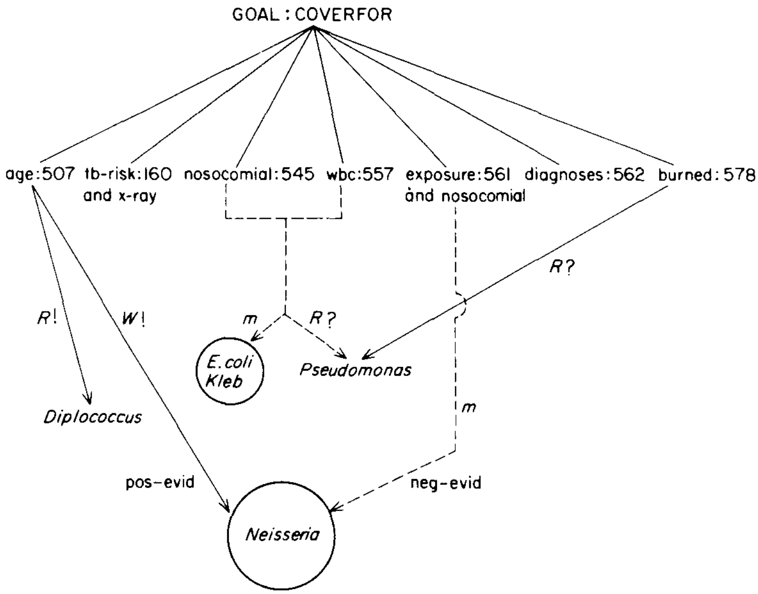

FIG. 9. Interpreting a student hypothesis in terms of expert rules. D-rules that conclude about organisms to "cover for" are shown with their key factors (see Fig. 6). Circled values are missing from the student's hypothesis (e.g. *E. coli*) or wrongly stated (e.g. *Neisseria*). Dotted lines lead from rules the student probably did not use. *m*, Evidence link that the tutor deduced is unknown to the student; *R* and *W*, links to right and wrong values that the tutor believes are known by the student; !, unique link; expert knows of no other evidence at this time; ?, questionable; tutor is not certain which evidence was considered by the student. *R*? means that the student stated this value; it is correct; and more than one d-rule supplies evidence for it.

(1) Variation in the premise of a d-rule: The student is using a d-rule that fails to apply or applies a successful d-rule prematurely (he is misinformed about case data or is confused about the d-rule's premise).
(2) Variation in the action of a d-rule: The student draws the wrong conclusion (wrong value and/or degree of certainty).

6.2. PRESENTATION METHODS FOR D-RULES THE STUDENT DID NOT CONSIDER

Returning to our example, after updating the overlay model, the tutor needs to deal with discrepancies between the student's hypothesis and what the expert program knows. The following t-rules are from a packet that determines how to present a d-rule that the student evidently did not consider. In our example, t-rule 9.02 generated the question shown in lines 11–17 (of Fig. 8). T-rule 9.03 (a default rule) generated the question shown in lines 19–43.

Apply the first tutorial rule that is appropriate:

T-RULE9.01

If: (1) The d-rule is not on the syllabus for this case, and
(2) Based on the overlay model, the student is ignorant about the d-rule
Then: Affirm the conclusions made by the d-rule by simply stating the key factors and values to be concluded

T-RULE9.02

If: The goal currently being discussed is a true/false parameter
Then: Generate a question about the d-rule using "facts" format in the premise part and "actual value" format in the action part

T-RULE9.03

If: True
Then: Generate a question about the d-rule using "fill-in" format in the premise part and "actual value" format in the action part

6.3. CHOOSING QUESTION FORMATS

When the tutor responds to a hypothesis, the context of the dialogue generally determines which question format is appropriate. However, during other dialogue situations it is not always clear which format to use (e.g. when quizzing the student about a rule that MYCIN has just applied using case data just given to the student). Our strategy is to apply special t-rules to determine which formats are logically valid for a given d-rule, and then choose randomly from the candidates.

T-rule 3.06 is part of a packet of t-rules that chooses an appropriate format for a question based on a given d-rule. The procedure for formatting a question is to choose templates for the action part and premise part that are compatible with each other and the d-rule itself.

T-RULE3.06

If: (1) The action part of the question is not "wrong value," and
(2) The action part of the question is not "multiple choice," and
(3) Not all of the factors in the premise of the d-rule are true/false parameters,
Then: Include "multiple choice" as a possible format for the premise part of the question

T-rule 3.06 says that if the program is going to give a conclusion that differs from that in the d-rule it is quizzing about, it shouldn't state the premise as a multiple choice. Also, it would be nonsensical to state both the premise and action in multiple choice form. (This would be a "matching" question—it is treated as another question type.) Clause 3 of this t-rule is necessary because it is nonsensical to make a multiple-choice question when the only choices are "true" and "false."

As can be seen here, the choice of a question type is based on purely logical properties of the rule and interactions among question formats. About 20 question types (combined premise/conclusion formats) are possible in the current implementation.

7. Final remarks

We have argued in this paper that it is desirable to add teaching expertise and other levels of domain knowledge to MYCIN-like expert programs if they are to be used as educational programs. Furthermore, it is advantageous to provide a flexible framework for experimenting with teaching strategies, for we do not know the best methods for presenting MYCIN-like rule bases to a student.

The framework of the GUIDON program includes knowledge of discourse patterns and the means for determining their applicability. The discourse patterns we have codified into procedures permit GUIDON to carry on a mixed-initiative, goal-directed case method dialogue in multiple domains. These patterns are invoked by tutoring rules, which are in turn controlled by a communication model. The components of this model are a case syllabus (topics the tutor plans to discuss), an overlay model (domain knowledge the tutor believes is being considered by the student), and a focus record (topics recently mentioned in the dialogue). Finally, we observed that meta-knowledge about the representation and use of domain rules made it possible to use these rules in a variety of ways duing the dialogue. This important because GUIDON's capability to flexibly reason about domain knowledge appears to be directly related to its capability to guide the dialogue in multiple, interesting ways.

Furthermore, we have augmented the performance knowledge of MYCIN-like systems by making use of support knowledge and meta-level abstractions in the dialogue. The problem-solving trace provided by the interpreter is augmented by GUIDON to enable it to plan dialogues (by looking ahead to see what knowledge is needed to solve the problem) and to carry on flexible dialogues (by being able to switch the discussion at any time to any portion of the AND/OR solution tree).

Development of GUIDON is still in its early stages. The procedures and rules described in this paper constitute the basic foundation of the program, but much experimentation remains to be done. We are just at the stage of trying the program with medical students; no formal experiments have been run at this time. The program has not been applied to all of the various EMYCIN domains, though we foresee no difficulties in this implementation. It remains to be seen just how domain independent the tutorial strategies are, or whether a mixed-initiative dialogue is even suitable for computational problems like that in the MARC domain of structural analysis.

Early experience with this program has shown that the tutor must be selective about its choice of topics if the dialogues are not to be overly tedious and complicated. That is, it is desirable for tutorial rules to exert a great deal of control over which discourse options are taken. Future development of GUIDON will focus on the use of the case syllabus for controlling the tutorial rules; we believe that it is chiefly in selection of topics and emphasis of discussion that the "intelligence" of this tutor resides.

The following people read earlier drafts of this paper and provided thoughtful remarks: Jan Aikins, Avron Barr, Jim Bennett, John Brown, Bruce Buchanan, and Adele Goldberg and Derek Sleeman. I would especially like to thank Bruce and Avron for allowing me to use portions of a working paper that we wrote together.

Many people have contributed their ideas to the MYCIN program over the past 4 years. The GUIDON project would not have been possible without their effort.

This reearch was sponsored in part by grants from ARPA (Contract Title MDA 903-77-C-0322) and NSF (MCS 77-02712).

References

BARR, A., BEARD, M. & ATKINSON, R. C. (1976). The computer as a tutorial laboratory: the Stanford BIP Project. *International Journal of Man-Machine Studies*, **8**, 567–596.

BOBROW, D. G., KAPLAN, R. M., KAY, M. *et al.* (1977). GUS, a frame-driven dialog system. *Artificial Intelligence*, **8** (2), 155–173.

BROWN, J. S. (1977). Uses of AI and advanced computer technology in education. In *Computers and Communications: Implications for Education.* New York: Academic Press.

BROWN, J. S., RUBENSTEIN, R. & BURTON, R. (1976). Reactive learning environment for computer-aided electronics instruction. *BBN 3314.*

BROWN, J. S., BURTON, R., MILLER, M., DE KLEER, J., PURCELL, S., HAUSMANN, C. & BOBROW, R. (1975). Steps toward a theoretic foundation for complex, knowledge-based CAI. *BBN 3135.*

BRUCE, B. C. (1975). Generation as a social action. In SCHANK, R. & NASH-WEBBER, B. L., Eds. *Theoretical Issues in Natural Language Processing.* pp. 74–77.

CARBONELL, J. R. & COLLINS, A. (1973). Natural semantics in Artificial Intelligence. *Proceedings of the 3rd IJCAI,* pp. 344–351.

CARR, B. & GOLDSTEIN, I. (1977). Overlays: a theory of modelling for CAI. *MIT AI Lab. Memo 406.*

COLLINS, A. (1976). Processes in acquiring knowledge. In ANDERSON, SPIRO & MONTAGUE, Eds, *Schooling and Acquisition of Knowledge.* Hillsdale, N.J.: Erlbaum Associates.

DAVIS, R. (1976). Applications of meta level knowledge to the construction, maintenance and use of large knowledge bases. *AIM-283.* Stanford University.

DAVIS, R., BUCHANAN, B. & SHORTLIFFE, E. H. (1977). Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence Journal,* **8** (1).

DAVIS, R. & BUCHANAN, B. G. (1977). Meta level knowledge: overview and applications. *Proceedings of the 5th IJCAI,* pp. 920–927.

DAVIS, R. & KING, J. J. (1977). An overview of production systems. In ELCOCK, W. E., *Machine Intelligence* **8**: *Machine Representations of Knowledge.* John Wiley.

DEUTSCH, B. G. (1974). The structure of task-oriented dialogs. *IEEE Symposium for Speech Recognition,* pp. 250–253.

FAUGHT, W. S. (1977). Motivation and intensionality in a computer simulation model. *AIM-305.* Stanford University.

FEIGENBAUM, E. A. (1977). The art of Artificial Intelligence: I. Themes and case studies of knowledge engineering. *Proceedings of the 5th IJCAI,* pp. 1014–1029.

FEIGENBAUM, E. A., BUCHANAN, B. G. & LEDERBERG, J. (1971). On generality and problem solving: a case study using the DENDRAL program. *Machine Intelligence,* **6**. Edinburgh University Press.

FEURZEIG, W., MUNTER, P., SWETS, J. & BREEN, M. (1964). Computer-aided teaching in medical diagnosis. *Journal of Medical Education,* **39**, 746–755.

GOLDSTEIN, I. (1977). The computer as coach: an athletic paradigm for intellectual education. *MIT AI Lab. Memo 389.*

GOLDSTEIN, I. & ROBERTS, B. R. (1977). Nudge, a knowledge-based scheduling program. *Proceedings of the 5th IJCAI,* pp. 257–263.

HARTLEY, J. R., SLEEMAN, D. H. & WOODS, P. (1972). Controlling the learning of diagnostic tasks. *International Journal of Man–Machine Studies,* **4**, 319–340.

HEISER, J. F., BROOKS, R. E. & BALLARD, J. P. (1978). Progress report: a computerized psychopharmacology advisor. *Proceedings of the 11th Collegium Internationale Neuro-Psychopharmacologicum,* Vienna, Austria.

KOFFMAN, E. B. & BLOUNT, S. E. (1973). Artificial Intelligence and automatic programming in CAI. *Proceedings of the 3rd IJCAI,* pp. 86–94.

LENAT, D. B. (1976). AM: an Artificial Intelligence approach to discovery in mathematics as heuristic search. *AIM-286.* Stanford University.

POPLE, H. E. (1977). The formation of composite hypotheses in diagnostic problem solving—an exercise in synthetic reasoning. *Proceedings of the 5th IJCAI,* pp. 1030–1037.

SCOTT, A. C., CLANCEY, W. J., DAVIS, R. & SHORTLIFFE, E. H. (1977). Explanation capabilities of production-based consultation systems. *American Journal of Computational Linguistics,* Microfiche 62.

SHORTLIFFE, E. H. (1974). MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection. *Ph.D. dissertation* in Medical Information Sciences, Stanford University. Also, *Computer-Based Medical Consultations: MYCIN.* New York: American Elsevier, 1976.

SHORTLIFFE, E. H. & BUCHANAN, B. G. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, **23**, 351–379.

SLEEMAN, D. (1977). A system which allows students to explore algorithms. *Proceedings of the 5th IJCAI*, pp. 780–786.

SPROSTY, P. J. (1963). The use of questions in the diagnostic problem solving process. In JACQUEZ, J. A., Ed., *The Diagnostic Process, Proceedings of a Conference Held at the University of Michigan*, pp. 281–308.

VAN MELLE, W. (1974). Would you like advice on another horn? Unpublished paper, Stanford University.

WEISS, S. M., KULIKOWSKI, C. A. & SAFIR, A. (1977). A model-based consultation program for the long term management of glaucoma. *Procceedings of the 5th IJCAI*, pp. 826–832.

WINOGRAD, T. (1977). A framework for understanding discourse. *AIM-297*. Stanford University.

YU, V. L., BUCHANAN, B. G., SHORTLIFFE, E. H., WRAITH, S. M., DAVIS, R., SCOTT, A. CARLISLE, & COHEN, S. N. (1978). Evaluating the performance of a computer-based consultant. To appear in *Computer Programs in Biomedicine*.