October 1981

Report. No. STAN-CS-81-894

Also numbered:
HPP-81-18

In W. Kintsch, J. Miller, and P. Polson (Eds.), Method and Tactics in Cognitive Science (pp. 51–83). Hillsdale, NJ: Lawrence Erlbaum Associates, 1984.
[Reprinted in G. P. Kearsley (Ed.), Artificial Intelligence and Instruction: Application and Methods (pp. 193–227). Reading, MA: Addison-Wesley, 1987.]

Methodology for Building An Intelligent Tutoring System

by

William J. Clancey

Department of Computer Science

Stanford University
Stanford, CA 94305

METHODOLOGY FOR BUILDING
AN INTELLIGENT TUTORING SYSTEM

William J. Clancey

Department of Computer Science
Stanford University, Stanford CA 94305

Contract No. N000C14-79-0302, effective March 15, 1979.
Expiration Date: March 14, 1979
Total Amount of Contract -- $396,325
Principal Investigator, Bruce G. Buchanan (415) 497-0935
Associate Investigator, William J. Clancey (415) 497-1997

Sponsored by:
Office of Naval Research,
Personnel and Training Research Programs,
Psychological Sciences Division.
Contract Authority No. NR 154-436
Scientific Officers: Dr. Marshall Farr and Dr. Henry Halff

The views and conclusions contained in this document are those of the authors and should not be interpret as necessarily representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. Government.

METHODOLOGY FOR BUILDING AN INTELLIGENT TUTORING SYSTEM

William J. Clancey

Heuristic Programming Project
Computer Science Department
Stanford University

1 Introduction

Over the past 5 years we have been developing a computer program to teach medical diagnosis.[1] Our research synthesizes and extends results in artificial intelligence (AI), medicine, and cognitive psychology. This paper describes the progression of the research, and explains how theories from these fields are combined in a computational model. The general *problem* has been to develop an "intelligent tutoring system" by adapting the MYCIN "expert system."[2] This conversion requires a deeper understanding of the nature of expertise and explanation than originally required for developing MYCIN, and a concomitant shift in perspective from simple performance goals to attaining psychological validity in the program's reasoning process.

Others have written extensively about the relation of artificial intelligence to cognitive science (e.g., [Pylyshyn, 1978] [Boden, 1977]). Our purpose here is not to

[1] The members of this project include the author, Research Associate in Computer Science at Stanford University, Bruce Buchanan, Adjunct Professor of Computer Science; two Research Assistants, Reed Letsinger (MS candidate in AI program) and Bob London (PhD candidate in the Department of Education); Dr. Timothy Beckett, Research Fellow at Stanford University Medical Center. Funding has been provided in part by ARPA and ONR Contract N00014-79C-0302. Computing resources are provided by the SUMEX-AIM facility (NIH Grant RR00785-07).

[2] A glossary appears at the end of this paper.

repeat those arguments, but to present a case study which will provide a common point for further discussion. To this end, to help evaluate the state of cognitive science, we will outline our methodology and survey what resources and viewpoints have helped our research. We will also discuss pitfalls that other AI-oriented cognitive scientists may encounter. Finally, we will present some questions coming out of our work which might suggest possible collaboration with other fields of research.

2 Goals: Intelligent Tutoring Systems

An *intelligent tutoring system* is a computer program that uses artificial intelligence techniques for representing knowledge and carrying on an interaction with a student [Brown & Sleeman, 1981]. Among the most well-known systems are WHY [Collins, 1976] (uses Socratic principles for teaching causal reasoning in domains like meteorology), SOPHIE [Brown, Burton, & Bell, 1974b] (provides a "simulated workbench" in which a student can test electronic troubleshooting skills), and WEST [Burton, 1979] (coaches a game-player on methods and strategies for exploiting game rules). This work derives from earlier efforts in computer-aided instruction, but differs in its attempt to use a principled or theoretical approach. First and foremost, this entails separating subject material from teaching method, as opposed to combining them in ad-hoc programs. By stating teaching methods explicitly, one gains the advantages of economical representation (the methods can be applied flexibly in many situations and even multiple problem domains) and the discipline of having to lay out subject material in a systematic, structured way, independently of how it is to be presented to the student. So the primary application of AI to these instructional systems is in the representation of teaching methods and domain knowledge. Ideally, this enterprise involves having a theory of teaching and the nature of the knowledge to be taught.

When we separate domain knowledge from the procedures that will use it, we say that we are representing knowledge "declaratively" [Winograd, 1975] (with respect to those procedures). *For example, in a medical domain, we would represent* links between data and diagnoses so they could be accessed and used for solving any given problem. A strong advantage of this approach is that the tutoring system can cope with arbitrary student behavior: no matter what order the student chooses to collect data (or troubleshoot a circuit, or make moves in a game), the program can evaluate partial solutions, and use its teaching knowledge to respond. Typically, the declaratively-stated knowledge base of diagnostic rules, causal relations, and the like is used during a tutorial to generate an "expert's solution," which, when compared to the student's behavior, provides a basis for *advising the student.*[3] The combination of a knowledge base of this kind and an interpreter for applying it to particular problems constitutes an *expert system*. So an intelligent tutoring system has an expert system inside it (Figure 1).

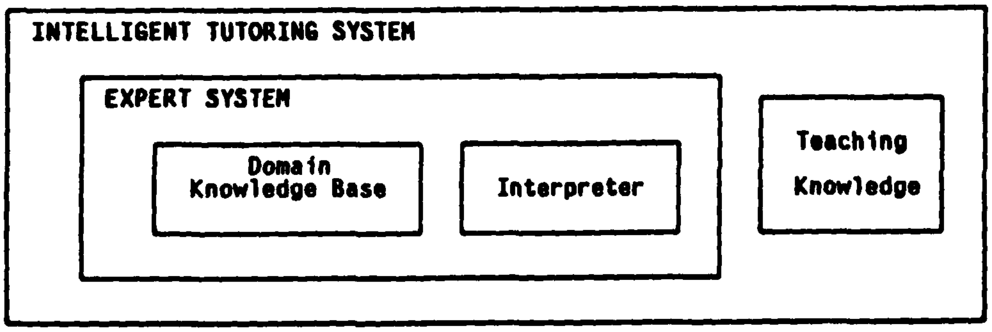

Figure 1. Components of an Intelligent Tutoring System

[3] In such a "first order" system, the model of the student's knowledge, as built by the program, is a subset of the internal, idealized knowledge base. This kind of model does not take into account student misconceptions or "bugs," an important area of research (see, *for example*, [Stevens, Collins, & Goldin, 1978] and [Brown & Burton, 1978]). The research described in this paper focuses on the (as yet unsolved) problem of constructing the expert knowledge base, the material to be taught.

In general, an "expert system" is a kind of AI program that is designed to provide advice about real world problems that require specialized training to master. Some examples are MYCIN [Shortliffe, 1976], which provides advice about antibiotics for infectious diseases; SU/X [Nii & Feigenbaum, 1978], which analyzes sonar signals; and R1 [McDermott, 1980], which configures the components of computer systems. These systems are built by interviewing experts in the given domain, and representing their knowledge in the *form of heuristics*, or "rules of thumb." For example, in an *expert system for field* biologists, one might find the rule, "if there are many buttercups and goldfield flowers, then the kind of underlying rock is probably serpentine." We call this kind of conditional statement, consisting of a premise and conclusion, a *production rule*.

In expert systems, there is no attempt to *simulate how* human experts think, for example to model the order in which they typically attack a problem. Instead, these programs are intended to capture the efficient leaps an expert makes from a problem description to an interpretation. This is what a production rule does. Expert systems differ in the nature of the task they solve (constructive, diagnostic, interpretative, etc.) and in their formalism for representing knowledge ("frames," "semantic nets"), but they all use rule-like associations.

The interpreter of an expert system is a program which controls the order in which the rules are considered. Common control strategies are *backward chaining* (working backwards from a goal) and *forward inferencing* (applying those rules whose conditions are satisfied by the problem description). These strategies correspond to two common ways of structuring the rule base, namely by the goal mentioned in the conclusion and by the problem description mentioned in the premise. By this structuring, the interpreter can index the rules and apply them. By the same token, the structure of a given rule base constrains how it can be used, the possible kinds of strategies the interpreter can use to access it.

The particular tutoring system we will be considering is built upon the knowledge of the MYCIN expert system. MYCIN's rules have to be restructured in order to be applied to teaching; the new system is called NEOMYCIN [Clancey and Letsinger, 1981]. Our methodology for building NEOMYCIN is the subject of this paper. The key idea is that using an expert system for teaching requires a shift in orientation from simply trying to output good solutions, to simulating in some degree of detail the reasoning process itself. The production rules that are used by MYCIN to provide good advice are inadequate for use as teaching material because certain kinds of reasoning steps, whose rationale needs to be conveyed to a student, are implicit in the rules. We need a more explicit, psychologically valid model of problem solving--one that can be understood and *remembered* by a student and incorporated in his behavior.

3 From MYCIN to GUIDON (an AI enterprise)

MYCIN is an expert system that was developed by a team of physicians and AI specialists. The program was designed to advise non-experts in the selection of antibiotic therapy for infectious diseases. The domain knowledge base (refer to Figure 1) contains approximately 450 rules which deal with diagnosis of bacteremia, meningitis and cystitis infections. The interpreter uses backward chaining, working from high order goals such as "determine whether the patient requires treatment" down to more specific subgoals such as "determine whether the patient has high risk for Tuberculosis." A typical rule is, roughly stated, "if the patient has been receiving steroids, then his risk for Tuberculosis meningitis is increased." Most rules are modified by a "Certainty Factor" indicating the rule author's degree of belief, on a scale from -10 to 10 that the conclusion holds when the premise is known to be true. Figure 2 shows excerpts from the diagnostic portion of a MYCIN consultation. Rules are chained together, working downwards from the high order goals; the program asks a question when it needs data to apply a rule. After the diagnosis is complete, a therapy program selects the most optimal therapy for the organisms most likely to be causing the infection. Additional tests might also be ordered.

```
--------PATIENT-1--------
1) Patient's name:
** John Smith
2) Age:
** 18 YEARS
3) Sex:
** MALE

     . . .

6) Are there any pending cultures for John Smith?
** YES

   --------CULTURE-1--------
   7) From what site was the specimen for CULTURE-1 taken?
   ** CSF
   8) Please give the date and time when the pending csf culture
      (CULTURE-1) was obtained. (mo/da/yr time)
   ** 8-Nov-88

     . . .

15) Has John Smith recently had symptoms of persistent headache or other
    abnormal neurologic symptoms (dizziness, lethargy, etc.)?
** YES

     . . .

22) Does John Smith have evidence of ocular nerve dysfunction?
** NO
23) Is John Smith a compromised host?
** NO

     . . .
```

Figure 2. Excerpt from MYCIN consultation

The success of MYCIN as a problem-solver, as measured in several formal evaluations [Yu, et al., 1979a][Yu, et al., 1979b], encouraged us to explore its application for teaching. The program's good performance, coupled with an ability to explain its line of reasoning, made it seem particularly suitable as teaching material. The rules had been acquired from physicians over many hours of discussion, comparing the program's behavior to their judgment, modifying rules to improve the program and testing the program on new problems. The rules pertaining to infectious meningitis were especially carefully constructed from experience with over 100 cases from local hospitals and medical journal articles. Therefore, we decided to focus on using the meningitis rules for teaching.

In order to understand what is good about MYCIN's rules and how they fall short for use in teaching, one must understand something about their construction, and what kind of explanation a tutorial program can provide by using them. Rules are not written independently of the whole rule base: a rule author must think about how a given rule will fit. Any given rule must make a conclusion about some goal that appears in at least one other rule premise, otherwise the rule would never be used (recall the mechanism of backward chaining). Moreover, some means must be provided to evaluate the subgoals mentioned in the premise, by writing other rules to make the appropriate conclusions and/or by making it possible for the system to gain the information from the user. So, in effect, a rule author is writing a kind of program in which goals are chained together by rules.

The author's choice of goals in the program constitutes a decomposition of the problem into reasoning steps. Figure 3 shows part of this internal goal structure in MYCIN. One method for determining the type of the infection brings into consideration whether the infection is meningitis and whether the patient has leukemia. To determine if the patient has leukemia, the program checks to see if the patient is immunosuppressed, and so on.

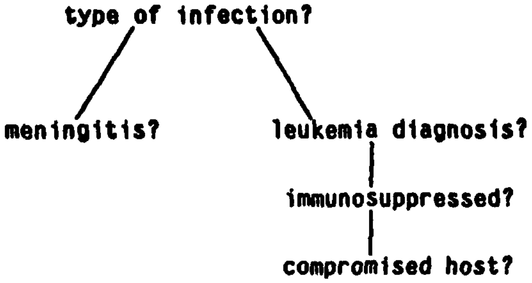

Figure 3. Portion of MYCIN's goal structure

The explanation capability of MYCIN [Scott, et al., 1977] is based upon the assumption that these steps, provided by a human expert, will make sense to the consultation program user. Figure 4 shows how the steps factor into an explanation of why MYCIN is requesting a datum. After the rule currently being considered is printed, repeated "Why" questions are answered by moving up the inference stack. Thus, the goals of Figure 3 are mentioned in the context of the particular rules now being considered that tie them together. This constitutes the program's "line of reasoning."

```
   23) Is John Smith a compromised host?
   ** WHY

[i.e. WHY is it important to determine whether PATIENT-1 is a compromised
     host?]

[3.0] This will aid in determining whether PATIENT-1 is immunosuppressed.
     If
         [3.1] John Smith is not a compromised host
     then
         it is definite (1.0) that John Smith is not immunosuppressed
     [RULE343]

   [back to question 23...]
   ** WHY

[i.e. WHY is it important to determine whether PATIENT-1 is immunosuppressed?]

[4.0]... in order to determine the diagnoses of PATIENT-1.
     If
         [4.1] John Smith is not immunosuppressed
     then
         it is definite (1.0) that leukemia is not one of the diagnoses of
             John Smith
         it is definite (1.0) that lymphoma is not one of the diagnoses of
             John Smith
     [RULE395]

   [back to question 23...]
   ** WHY

[i.e. WHY is it important to determine the diagnoses of PATIENT-1?]

[5.0]... in order to determine the type of the infection.
     It has already been established that
         [5.1] the infection which requires therapy is meningitis, and
         [5.2] only circumstantial evidence is available for this case, and
         [5.3] John Smith does not have a tb risk factor
     Therefore, if
         [5.4] 1) Leukemia is one of the diagnoses of John Smith, or
               2) Lymphoma is one of the diagnoses of John Smith
     then
         there is weakly suggestive evidence (.3) that the type of the
             infection is not tb
     [RULE056]

   [back to question 23...]
   ** NO
```

Figure 4. Excerpt of MYCIN consultation showing explanation capability (User input follows **.)

In the "question-answer" interaction after MYCIN has given its advice, the user can inquire in more detail about how data was used and how conclusions were made (Figure 5). Again, all explanations are based on the rules and goals of the system.[4]

```
** DID YOU CONSIDER WHETHER THE PATIENT WAS AN ALCOHOLIC?

RULE535 could have used:
     whether John Smith is an alcoholic

Clause 5 of RULE535 {the patient is an alcoholic} was already known to be
     false for CULTURE-1.

** HOW DID YOU KNOW THAT THE PATIENT WAS NOT AN ALCOHOLIC?

RULE343 was used to conclude that John Smith is not an alcoholic.  The last
     question asked before the conclusion was made was 23.

** WHAT WAS QUESTION 23?

Question 23 was asked in order to find out whether John Smith is a
     compromised host in an effort to execute RULE343.
```

Figure 5. Excerpt of question/answer interaction after a consultation

There are two important kinds of explanations that MYCIN cannot give: *it cannot explain why a particular rule is correct, and it cannot explain the strategy behind the design of its goal structure*. These deficiencies only became important to us in the course of developing GUIDON. In effect, we were forced to reconsider our conception of "transfer the expertise," the model by which we viewed the process of representing expertise and using it in an AI program (Figure 6).

[4] The AI technology that makes this possible--giving the program knowledge about its representation so that it can dissect its rules--is not of prime concern to us here. The interested reader can find details in [Davis, 1976].

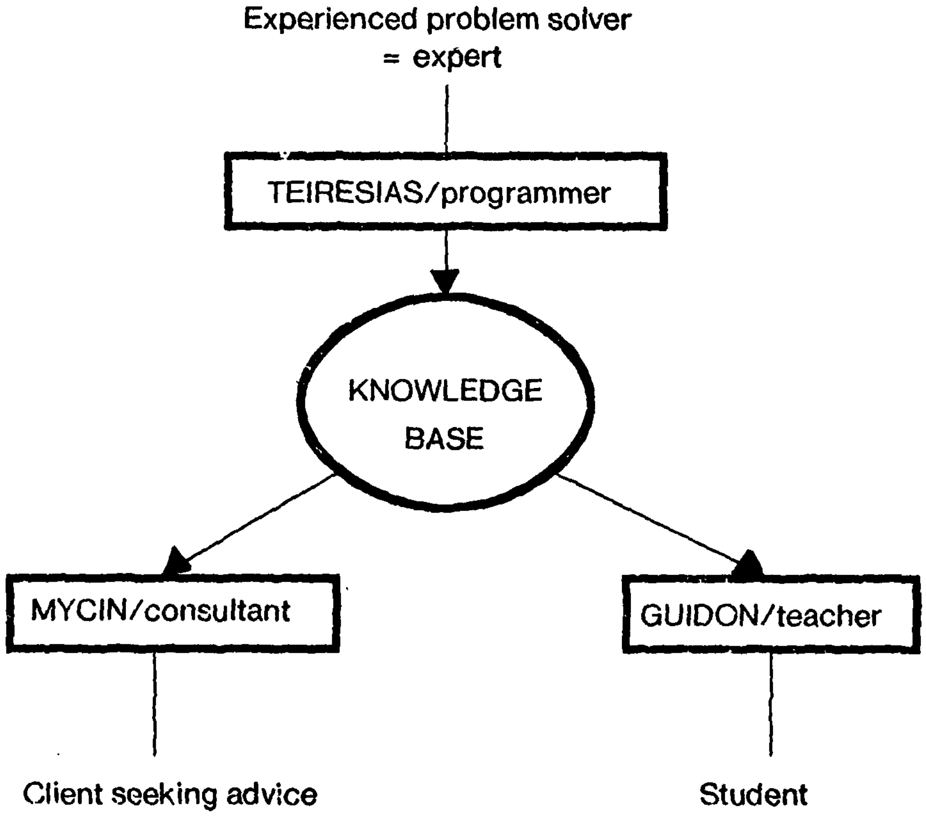

Figure 6. Transfer of expertise: Learning, Advising, and Teaching

Instead of thinking of expertise transfer as a process of moving knowledge down a pipeline from an expert (via Davis' Teiresias system [Davis, 1979]) to a client or student, we were forced to be more precise about the different kinds and purposes of knowledge. At a certain level, MYCIN is aphasic--able to perform, but unable to talk about what it *knows*. Teaching and explanation, we came to *recognize*, place different demands on an expert than simply solving problems. A teacher can provide analogies, multiple views, and levels of explanation which are unknown to MYCIN. In building MYCIN, we did not make explicit ***how an expert organizes his knowledge, how he remembers it, and strategies he uses for approaching problems.***

These shortcomings became clear in the course of developing *GUIDON, a project* undertaken in a typical AI way as a "constructive experiment." GUIDON was a test of the assumption that the rule base could be used for purposes other than problem-solving, so in designing the system we held the rules fixed. Moreover, to test the idea that teaching knowledge could be represented wholly separate from domain expertise, the system was designed to work with any rule base written in MYCIN's language (see [van Melle, 1980]). Therefore, the rules are treated syntactically: GUIDON's teaching procedures are designed *to discuss how* a goal is achieved in terms of the data appearing in rule premises and to evaluate a student's hypotheses in terms of *conclusions made by* the rules. Figure 7 shows an excerpt from a dialogue with GUIDON.[5]

[5] The teaching procedures are not our main concern here. See [Clancey, 1979a] [Clancey, 1979b] for details.

```
** FACTORS

The following factors will be useful to determine the type of the infection:

                 3a.  whether the meningitis is partially treated
                 3b.  whether a lumbar puncture has been performed on John Smith
                 3c.  the duration of the neurological signs
                 3d.  the CSF protein
             (*) 3e.  whether John Smith has symptoms of Herpes Zoster virus
                                         .
                                         .    {Section omitted}
                                         .

  Factors marked with '*' have already been stated.  Based on what we have
  already mentioned, you should be able to conclude about: 3b.

** USE 3C

The duration of the neurological signs is 7.2 hours.

Using other information available to you now, the duration of the
neurological signs enables you to conclude that the type of the infection is
bacterial (.2) viral (.2) [RULE524].


** IKNOW 3B

Good.  MYCIN has also made a decision.

The site of CULTURE-1 is evidence that a lumbar puncture has been performed
on John Smith [RULE112].

Back to our discussion of the type of the infection....

** 3D

The CSF protein is 40.

** HYPOTHESIS VIRAL

You're right, there's strong evidence that the type of the infection is
viral, given the CSF protein, the duration of neurological signs, and the
fact that the patient has symptoms of Herpes Zoster virus.

What other facts about this case tell you that the type of the infection
is viral?
(Enter one factor per line.)
>>
                                          .
                                          .
                                          .
```

Figure 7. Excerpt of GUIDON Interaction

GUIDON was improved over time by annotating the rules to separate the "key factors" of the rule from contextual and "screening" clauses that determine applicability of the rule (so complex rules with 5 clauses and a table of conclusions can be summarized by saying "the CSF protein is evidence for viral infection"). Clause distinctions of this kind are part of the implicit design knowledge that is unknown to MYCIN because it lies outside of the rule syntax.

But simple annotations were insufficient; the knowledge base also lacked medical knowledge necessary for teaching. We found that students were unable to remember the rules, even after discussing a single problem with GUIDON many times. Students who apparently knew what data to collect were unsure of the order in which to collect it, and consequently had no confidence that their investigations were complete. This experience suggested that the program needed to teach a problem-solving strategy that a student could follow, as well as some underlying mnemonic structure for understanding and remembering the rules. No formal experimentation was necessary, the program plainly lacked the necessary medical knowledge.

4 From GUIDON to NEOMYCIN (a Cognitive Science enterprise)

In the course of studying the teaching problem, we learned that the expertise and explanations of MYCIN are narrowly conceived. On the one hand, we have not captured all that an expert knows, for example, his causal models of disease processes by which he understands rules and is able to reason about when they can be violated. On the other hand, some of what we have captured is *implicit* in the rules, namely the taxonomic structure of diseases and the search strategy (top-down refinement). This knowledge is procedurally embedded in the choice of subgoals and their ordering in a rule. This is illustrated by the alcoholic rule (Figure 8).

```
RULE535
-------

  If:  1) The infection which requires therapy is meningitis,
       2) Only circumstantial evidence is available for this case,
       3) The type of the infection is bacterial,
       4) The age of the patient is greater than 17 years,
       5) The patient is an alcoholic,

  Then:  There is evidence that the organisms which might be
         causing the infection are diplococcus-pneumoniae (.3) and
         e.coli (.2).
```

Figure 8. The Alcoholic Rule.

Clauses 1-3 specify that this rule about particular bacteria will only applied after bacterial meningitis infection is established (three levels of the taxonomic hierarchy). Clause 4 is based on the fact that children are usually not alcoholics, illustrating that the rules are based on implicit knowledge about the world, too. Finally, the rationale for associating alcoholics with the listed bacteria is not represented. Figure 9 illustrates the different kinds of knowledge that the human expert relied upon to construct this rule, which we did not represent explicitly in the program. The kinds of knowledge are labeled as strategic, structural and support knowledge.

The MYCIN program shows us clearly that the task orientation to develop a program with a high level of performance alone does not lead to a process model of human problem solving. MYCIN does subgoaling, as people sometimes do, but it doesn't do diagnosis like people. For one reason, subgoaling is not the key element of diagnostic rule application; focussed forward-inferencing is. For teaching purposes, we need to model how an expert uses and remembers his knowledge--not just capturing the associations he makes, but capturing also why these associations come to mind. It is the task orientation of tutoring that makes these considerations relevant and that will be the measure of adequacy for the models we construct.

To recap, in building an intelligent tutoring system, we are forced to move beyond the constraints of performance and consider the psychological constraints of teaching. We need to be able to articulate how the rules fit together, how they are constructed. We have studied MYCIN's rules and developed an epistemology of the kinds of knowledge that relate to teaching of heuristics (Figure 9) [Clancey, forthcoming]. Following the theory, a new representation was developed in which the original MYCIN rule set is reconfigured to make these kinds of knowledge explicit [Clancey and Letsinger, 1981]. Figure 10 illustrates the main components of this new system, NEOMYCIN. With its theoretical, epistemological underpinning, NEOMYCIN is designed to represent the subject material that a new version of GUIDON can use to articulate important teaching points.

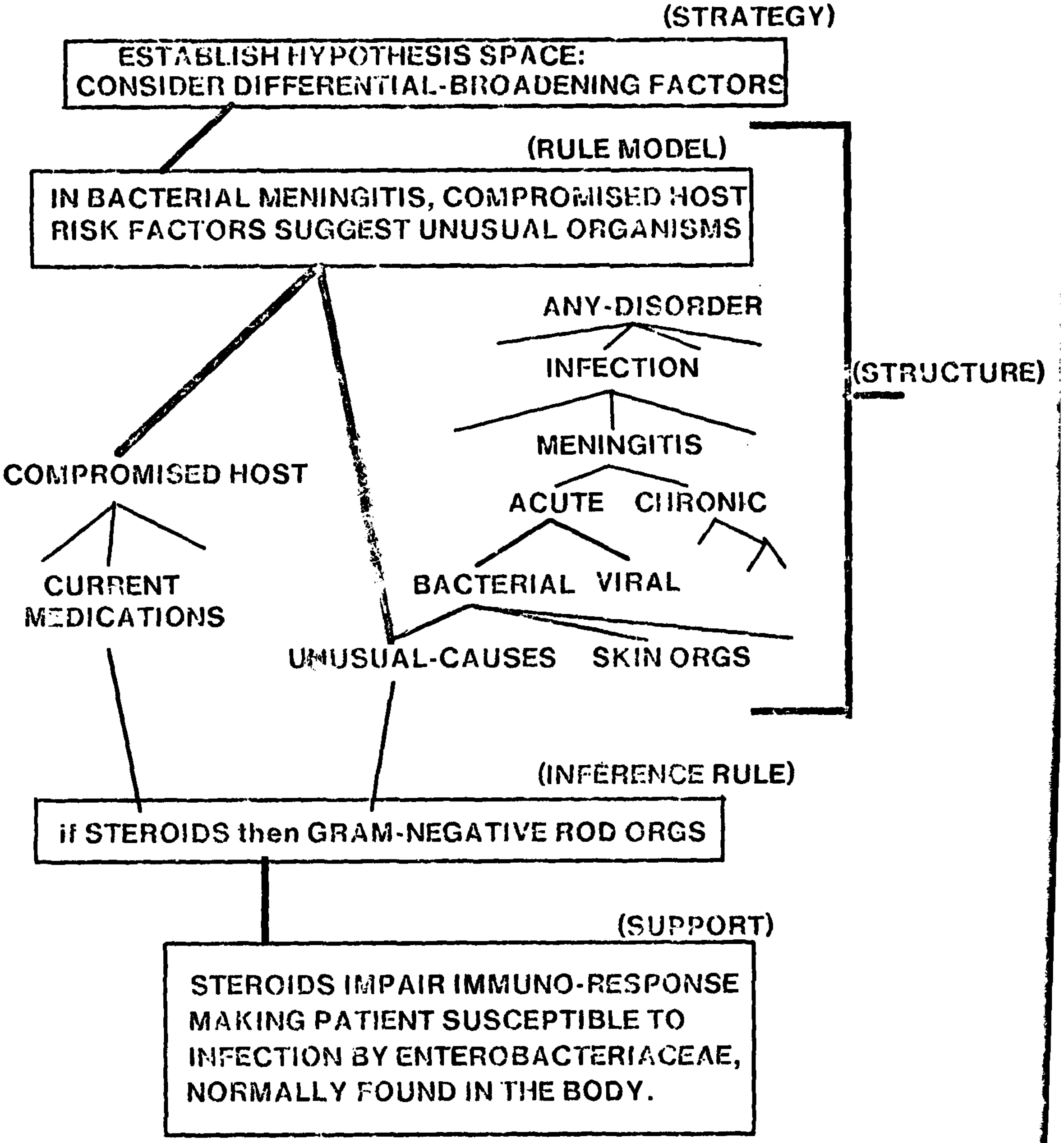
(STRATEGY)
ESTABLISH HYPOTHESIS SPACE:
CONSIDER DIFFERENTIAL-BROADENING FACTORS
(RULE MODEL)
IN BACTERIAL MENINGITIS, COMPROMISED HOST
RISK FACTORS SUGGEST UNUSUAL ORGANISMS
ANY-DISORDER
INFECTION
(STRUCTURE)
MENINGITIS
COMPROMISED HOST
ACUTE
CHRONIC
CURRENT
MEDICATIONS
BACTERIAL
VIRAL
UNUSUAL-CAUSES
SKIN ORGS
(INFERENCE RULE)
if STEROIDS then GRAM-NEGATIVE ROD ORGS
(SUPPORT)
STEROIDS IMPAIR IMMUNO-RESPONSE
MAKING PATIENT SUSCEPTIBLE TO
INFECTION BY ENTEROBACTERIACEAE,
NORMALLY FOUND IN THE BODY.

Figure 9. Strategic, structural, and supporting knowledge for a heuristic

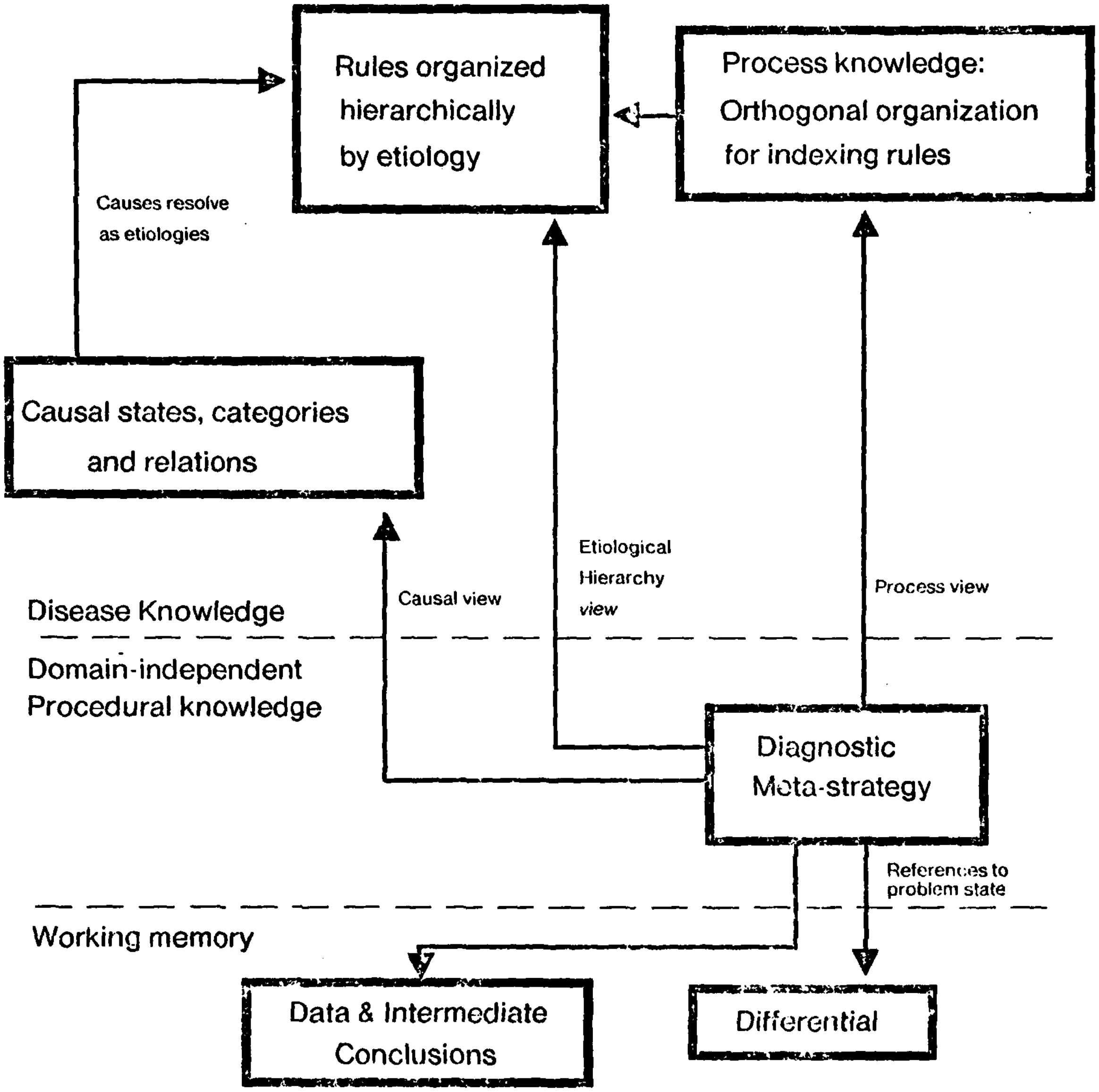
Rules organized hierarchically by etiology
Process knowledge: Orthogonal organization for indexing rules
Causes resolve as etiologies
Causal states, categories and relations
Etiological Hierarchy view
Causal view
Process view
Disease Knowledge
Domain-independent Procedural knowledge
Diagnostic Meta-strategy
References to problem state
Working memory
Data & Intermediate Conclusions
Differential

Figure 10. Components of the NEOMYCIN expert system.

Figure 10 shows that the key feature of NEOMYCIN is separation of domain-specific disease knowledge from general procedures for doing diagnosis. The strategical knowledge "gets a handle on" the disease knowledge by way of alternate "views" or structural organizations of the disease knowledge. It is through these general indexing relationships, such as the hierarchical relationships of "sibling" and "father," that a general procedure can examine and select specific problem-solving knowledge to apply it to a given problem. The causal view indexes the disease hierarchy through causal abstractions. (For example, "double vision" might be caused by "increased pressure in the brain" which might be caused by "a brain tumor," "a brain hemorrhage," and so on.) The process view pertains to general features of any disease which describe its location, progression of symptoms, degree of spread, and so on--general concepts by which the problem solver can index his knowledge about diseases to compare and contrast competing hypotheses. Figure 10 also shows the working memory which will be described later.

So far we have been considering how NEOMYCIN, as a representation, adheres to an epistemological theory of knowledge, that is, it separates out expertise by the divisions suggested in Figure 9. The "content" of NEOMYCIN is a psychological theory for gathering and interpreting new data, in part, the content of the meta-strategy box in Figure 10. NEOMYCIN embodies a psychological theory of medical diagnostic reasoning for the purpose of monitoring a student's problem solving and providing assistance that a student can follow. For example, we will be teaching forward-directed inferences--leaps from data to hypotheses--that we represent in NEOMYCIN's *trigger rules*. With this additional knowledge of how people think, GUIDON version 2 will have leverage for interrupting the student to test his knowledge, as well as having a better basis for understanding a student's partial solutions.

While this is not primarily a paper about teaching strategy, we hasten to clarify that we do not propose to directly teach students a model of what experts do. Indeed, the epistemological separation of knowledge in NEOMYCIN brings out individual steps of reasoning that we believe are "compiled" in experienced problem-solvers, just as in MYCIN's original rules. The point of the decomposition is to provide a rationale for surface expert behavior so a student can understand it. Thus, on the surface NEOMYCIN is designed to behave like an expert in its focussing, data collection and hypothesis formation. Moreover, the types and organization of knowledge are those of an expert. But the process itself is drawn out here into "diagnostic tasks" (the meta-strategy) that we believe an expert follows when stuck, but generally does not consciously consider, knowing what to do in each situation from years of practice.

Furthermore, we have not specified how this material will be presented to a student. The sequencing of material and various support stories for understanding and memory are part of the theory of teaching which we do not address here.

5 The Relation of Theory, AI Formalism and Program

NEOMYCIN is more than an ordinary AI system built to simply do some task. It is not an ad hoc system built to get performance--it is an implementation of a theory of diagnosis and certain principles for representing knowledge. Our tutoring goals require that the program combine both a theoretical model of medical diagnosis, so that the student's problem solving can be interpreted and advice offered, and an epistemological theory of knowledge, so that this model of diagnosis can be articulated to the student. These theories are instantiated in a program by way of AI formalisms for representing and controlling knowledge, some of which are novel and grew out of the theoretical goals. Figure 11 shows how theory and model are related in NEOMYCIN. This section will describe in more detail how the theories factor into the AI formalisms and the actual code of the system.

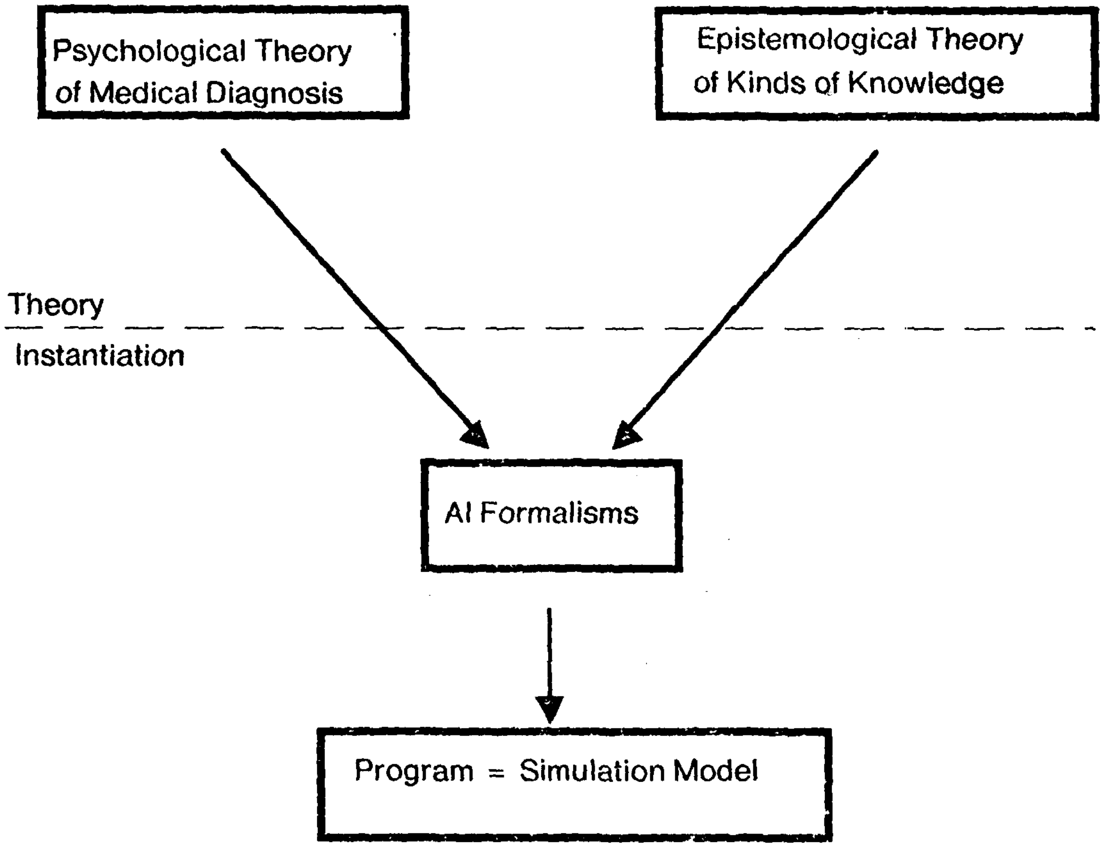
Psychological Theory of Medical Diagnosis
Epistemological Theory of Kinds of Knowledge
Theory
Instantiation
AI Formalisms
Program = Simulation Model

Figure 11. Relation of Theory to AI Formalisms and the Model.

5.1 The Psychological Theory of Medical Diagnosis

The questions addressed by the theory of medical diagnosis we are developing are: How does a physician use problem data and disease knowledge to formulate hypotheses, to request additional data, and to reach a diagnosis? Issues pertaining to the processing of new information, the structure of disease knowledge, the nature of procedural knowledge and its relation to disease knowledge, among others, are appropriate. The theory is general, both in its application to multiple problems in a given domain and its potential applicability to other domains, thus the problem of arbitrariness in process models [Greeno & Brown, forthcoming] is partially ameliorated. Underlying regularities become manifest through this constant consideration of multiple tasks and multiple domains. [Kosslyn, 1980][6]

The theoretical features described below do not literally appear anywhere in the program. These are descriptions of behavior that were written down clearly and explicitly before any coding began. They were not extracted from the program; they were designed into it.

In writing down the principles of the theory, we were almost always thinking about their implementation in the program, often requiring that we return to be more precise about the theory. For example, we could not simply write down that "data are used immediately in a forward-directed way." Should every rule that uses new information be allowed to fire? This did not fit our observations. For example, when thinking about "steroids" in the context of a possible meningitis case, inferences are obviously focussed by the problem at

[6] This is on top of the principled character of representation deriving from our epistemological framework. Generality stemmed first of all from our need for teaching general principles to students. Ultimately, the enterprise has engineering value: we can lift the representation framework as well as the domain-independent diagnostic strategy into another problem domain and develop a new consultation system with this as a starting point.

hand; tuberculosis might come to mind, not the possibility of a law suit in college athletics. In turn, our evolving knowledge representation (also on paper) suggested that this focussing might be modelled by only firing rules that appear in subtrees of the etiological hierarchy below hypotheses currently being entertained (so "tuberculosis" would come to mind because it is below "meningitis"). [7]

The following is a brief presentation of the key theoretical features of NEOMYCIN, fairly similar to how they appeared before we wrote any code:

a. Incoming data are immediately applied by forward-directed reasoning leading to more abstract descriptions of the problem and support for specific diagnostic hypotheses.

1) Trigger rules place hypotheses on the differential (working memory of hypotheses) directly as data is received. The differential is maintained so more specific causes replace general hypotheses.

2) Data are abstracted immediately, e.g., "diplopia" is thought of as a "abnormal neurological finding."

3) Process-oriented questions are immediately asked, relevant to the domain, but not directed to any particular hypothesis, e.g., asking when a symptom began and how it has changed over time.

4) Data suggest causal state-categories, possibly jumping over a chain of causal links to conjecture some generic problem whose subtypes

[7] This idea bears some obvious relation to frame theory; an elaboration is beyond the purpose of this paper.

are later considered (as "brain pressure" suggests "space-occupying substance in the brain" rather than the specific causes of "brain pressure).

5) Data/hypothesis associations are applied in the context of the current differential. Only associations that appear in subtrees below the current hypotheses come to mind.

b. The following knowledge sources are represented separately and explicitly, in accord with the epistemological theory:

1) a problem-space hierarchy to which data/hypothesis rules are attached ("etiological taxonomy") (previously implicit as the "context clauses" of rules);

2) causal rules that ultimately tie into this hierarchy (see Figure 10);

3) world relations that constrain the relevance of data (previously implemented as "screening clauses");

4) disease process knowledge that cuts across the etiological distinctions, useful for initial problem formulation.

c. A hierarchical set of domain-independent meta-rules constitute a diagnostic meta-strategy. These rules examine the knowledge sources listed above and the current differential to select an hypothesis to focus on and the next datum to request.

Turning now to the content of the strategic rules, we determined that the key strategic idea to teach students is that the purpose of collecting circumstantial evidence, in preparation for making physical measurements, *is to "establish the hypothesis space,"* to determine the *range of possibilities* that might be causing the problem. Strategies for achieving this involve looking for evidence that will broaden the space of possibilities by considering common and unusual causes.

There are two orientations when establishing the hypothesis space: 1) "group and differentiate"--upward-looking, initial problem formulation in which one tries to cluster the data under some generic process (cause); and 2) "explore and refine"--attempting to confirm successively more specific causes. The trigger associations mentioned above bring the problem solver "into the middle" of his problem space hierarchy. These strategies together establish a path to a diagnosis.

The initial problem formulation we want to teach goes beyond MYCIN's expertise, requiring both the strategy of "group and differentiate" as well as additional medical knowledge. Essentially, we want to teach a student not just how to confirm that meningitis is present, MYCIN's task, but *when one should think* about meningitis, and what it might be confused with. One normally associates these questions with the "primary care" physician, as opposed to a consultant like MYCIN. These perspectives, stemming from our tutorial goals, led us to adopt a more theoretical understanding of the *task* of diagnosis itself.

5.2 The AI Formalisms of NEOMYCIN

A tacit principle of AI is that an AI program must be describable in terms of theoretical formalisms of knowledge representation and control. Thus, in a real sense we might move the "theory/instantiation" line of Figure 11 to below the "AI formalisms" box. For just as what we write down about trigger rules in our psychological theory is separable from its implementation as code, the mathematical, logical and AI concepts of "antecedent rule," "hierarchy" and the like are abstractions for entities and processes in our FORTRAN or INTERLISP programs. However, they are apparently "closer" to our code than is the psychological theory, often even designated by procedure and variable names that make the correspondence explicit to the programmer.

A good example of the use of AI technology in NEOMYCIN are the diagnostic strategies which are represented as meta-rules, an adaption of a pre-existing formalism [Davis, 1976]. These rules are applied as a *pure-production system* for each subtask (e.g., "find a new focus" is a subtask). *Abort conditions* are inherited to simulate shifting of focus (and return to higher goals) as data broadens the differential or exploration suggests that a conjecture is unlikely.

We mention these examples of AI formalisms in NEOMYCIN to illustrate the point that a cognitive scientist doesn't simply sit down and write any program whatsoever as a model of his theory. As in mathematics and logic, there are certain notations that have been developed for couching theoretical relations, and the notations evolve as the theories become more complex. The work of writing AI programs is made much easier by previous efforts to abstract representational devices such as "meta-rules." These devices become like a bag of tools for expressing theories. In order to communicate the NEOMYCIN model to other AI programmers, it was essential to adapt whatever tools were already in common use, rather than inventing new terms or arbitrarily combining old formalisms. So in describing NEOMYCIN, indeed in *thinking about it*, we say that the meta-rules are applied as in a pure production system; the disease process knowledge is represented as a frame associated with each disease; and so on. Furthermore, AI's bag of tools provided a ready-at-hand, suggestive set of organizational and processing concepts for expressing the psychological theory. Finally, in this special case AI provided the data (MYCIN's rules) that enabled us to study human knowledge in a new way.

5.3 When is a Program Ad Hoc?

The scheme shown in Figure 11 provides an interesting handle on the question of ad hocness in computational models. It shows that there are multiple perspectives from which the model can be said to be ad hoc. From the AI perspective, code is ad hoc if it is loosely put together without regard for unified, simple and elegant formalisms. If NEOMYCIN's diagnostic meta-strategy had been implemented in INTERLISP procedures directly, instead of a hierarchy of meta-rules applied cyclically with abort conditions, etc., the implementation would be said to be ad hoc. Here ad hocness would have interfered with our teaching goals as well as program maintenance.

Moving up a level, if we had used MYCIN's rule language, an AI formalism, instead of the NEOMYCIN scheme of an etiological hierarchy combined with meta-rules, etc., our implementation would have been said to be atheoretic from the epistemological perspective. That is, we would have represented different kinds of knowledge in a uniform way, losing distinctions--in some sense of the essence of an ad hoc implementation. [8] Indeed, it was the ad hoc representation of strategies and taxonomic concepts in MYCIN rules that limited its usefulness for teaching.

[8] Notice the tension between the epistemological and AI formalism levels: without uniformity there is no formalism, but the uniformity chosen may not allow important distinctions to be expressed.

Finally, looking at the left theoretical arm of Figure 11, an implementation can be ad hoc from the psychological perspective. If we had persisted in using exhaustive, top-down refinement, as in MYCIN, and several other *medical* AI systems, we would have constructed a program that does medical reasoning, but in an ad hoc way, limiting the usefulness of the program for interpreting student behavior. Note that exhaustive top-down refinement is not an ad hoc implementation *from* the AI formalism perspective, but it is a psychologically implausible model of search.

Observe that from all three perspectives, it is the task orientation that *determined* what aspects of the implementation were relevant, those which should not be done in an ad hoc way. In general, the question of where we should "draw the line between implementation detail and relevant model content" [Kintsch, et al., *forthcoming*] depends on what we want to model, what we want the program to do. The attempt to apply the model to a *real* world task will provide the empirical feedback that reveals what was ad hoc and now needs to be implemented in a *theoretical way.* But *note* again, we do not *extract* the theoretical principles from the program (contrast with [Pylyshyn, 1978]) we write them down, then build them in. By default all other coding decisions will be ad hoc, and we won't know whether that matters *until we do more* testing.

5.4 Summary of NEOMYCIN as a Model

To summarize, NEOMYCIN is an information processing model which uses AI formalisms to instantiate psychological and epistemological theories of knowledge and processing:

a. The epistemological theory specifies how different kinds of knowledge interact, specifically how organizational knowledge interacts with strategies

b. What the expert does is not simply listed: the strategies are domain-independent, they specify how different kinds of knowledge sources are called into play to massage a guess that is being constructed and refined (domain-independence makes the process model more psychologically plausible and extensible [Greeno & Brown, forthcoming]).

c. Associations of data with hypotheses are described in terms of the working memory and a structured representation of the problem space (following from the diagnostic theory).

d. The model of strategies specifies hierarchical organization of knowledge in the form of rules for achieving tasks; the problem-solver is said to be oriented to "what he is trying to do" (diagnostic theory).

e. different kinds of follow-up questions are not simply listed: the model specifies how subgoals can be set up by associations that trigger when data is received, and how immediate follow-up questions are associated with data abstractions (diagnostic theory).

In short, NEOMYCIN specifies organization of different kinds of knowledge and processes by which this knowledge is called into play. It is a model that relates a working memory to the kinds of associations people try to think about and why they remember them at particular times. The overall theory is complex; the computer program provides a practical means of testing the coherency and completeness of the theory. [9]

6 Methodology

From the period February 1 through December 1, 1980 we met regularly with a

[9] See [Kosslyn, 1980] for further discussion of the relation of theories to programs as models.

physician consultant with the purpose of revising MYCIN's rules to make the teaching points clear. Protocol analysis (using cases MYCIN had previously solved) was the chief method. We also attended classes taught by this physician and compared them to another physician's handling of the same course. In addition, we presented several cases to our physician's best student to compare his reasoning and explanations to his teacher's.

Our physician's approach was logical and easy to emulate. After listening to several other physicians and sitting in on other classes, we decided that we had found an unusually good teacher, someone who was consistent from case to case, and moreover did what he told students to do. Other teachers we observed were not able to articulate their approach as clearly and seemed to be less sure of what students were thinking. There were common strategical concepts, however, that our experts all used to explain their reasoning ("hit the high points," "consider risk factors"). In our opinion, the reason our physician was a good teacher was because his explanations were not as "flat" as other physicians'. Rather than saying, "Well, the patient hasn't traveled, so it isn't Valley fever," he would say "Well, travel would have widened the spectrum of possibilities, so we can rule out things like TB picked up in Mexico." That is, he supplied abstractions that said *what he was trying to do*, how his thinking was oriented.

Our framework of structural, support, and strategic knowledge for organizing, justifying and controlling the use of heuristic rules served well in knowledge acquisition dialogues. We would always ask ourselves, "What kind of explanation is he giving us? A data/ hypothesis rule? Why he believes a rule? Why he thought to consider that association (the indexing, the approach)?" We organized these kinds of knowledge around each rule we discussed (Figure 9), and directed the conversation appropriately. In contrast, several years ago, before deriving this framework, our interviews tended to take a depth-first plunge into pathophysiological details (we always asked, "And what causes that?"), which did not shed much light on the physician's strategies and organization of data/hypothesis relations.

We tape-recorded sessions whenever a case was presented to the physician. A note file was maintained in which we recorded what we learned from each meeting. A summary of the kinds of interactions is given below (in the order they occurred).

A. *Informal discussion of a case previously diagnosed by MYCIN.* The experimenter presents data and asks how it is useful. Among the points of discussion: how the expert cuts up the problem (for example, acute vs. chronic), how he remembers data/hypothesis relations (diagnostic values are related to a mnemonic story), the significance of frequently-mentioned problem features ("predisposition," "compromised host"), how urgency and faulty data factor into reasoning. Later comparison of the expert's terms and rules to MYCIN's suggest questions for subsequent meetings.

B. *The expert solves a case, while the experimenter actively questions his reasoning throughout.* Initial data is presented, the expert must then request information in any order he desires, and make a diagnosis. Among the items we record: the differential (hypotheses under consideration); strategy (either a domain-specific goal, such as "look for evidence of a focal lesion," or a domain-independent goal, such as "pursue most likely causes first"[10]); rule-like associations ("diplopia suggests increased brain pressure"); and meta-statements about strategy ("think before the lab results, not from the results"; "make

[10] These are often stated aphoristically as well: "When you hear hoof beats, think of horses, not zebras."

reliability checks of data"). NEOMYCIN's strategy rules were first derived from analysis of one of these protocols.

Designing a general program from a single, typical interaction is a common method in AI. The knowledge base designer idealizes the interaction, specifying knowledge (frames, rules, etc.) and processing (general procedures, strategy rules) which will bring about the desired program behavior in the particular case. GUIDON's tutorial procedures were first sketched out in this way by proceeding from a sample interaction in which we played both the part of the student and teacher (generating realistic student input and then looking in the rule base to find what response would be satisfying). The program is generalized and debugged by testing it on many other cases afterwards. For example, single statements might become separate procedures as the complexity of problem situations becomes better understood. This method presupposes that the general framework of the system (or meta-strategy in the case of NEOMYCIN) can be induced, at least in preliminary form, from any particular problem solution.

C. *The expert is asked to describe a typical case for each of the main diagnoses.* The expert finds this easy to do. This method brings out the diagnostic or invariant associations, as well as what evidence is required to rule-out competing hypotheses. For comparison, the expert is asked to describe atypical presentations of the same disorders. (In these cases, the expert gives the impression that he is telling a joke.) From these analyses, we developed a theory of what makes a case easy or difficult. [11]

[11] Diagnosticity (sharpness of measurements (classicality), presence of important factors (but not necessary), presence of invariant factors (sufficient); dissonance (absence of extraneous (unexplained) factors) and inconsistencies (unexpected factors)); the a priori likelihood of the problem (expert has less confidence in unlikely diagnoses); and multiplicity of cause (before reaching a diagnosis, the expert will struggle to find a simpler, single explanation).

D. ***The expert is asked to present a case to the experimenter, reversing the roles of method B.*** This helps the experimenter determine whether he has formalized an "executable procedure." This method quickly reveals any gaps in knowledge or approach that have not been extracted from the expert. The expert is asked to present both easy and difficult cases so the evolving model can be more fairly evaluated.

E. ***The same cases discussed in B and D are presented to different experts.*** Since *we already understand* the significance of the data (the data/hypothesis rules) we are especially interested in comparing strategies which bring the data to mind.

F. ***The developed strategy model is presented to the original expert for his evaluation.*** What resonates with his thinking? What does he care to elaborate upon? Where do students have problems? (*The expert says* things like, "Most students encounter roadblocks--they're not sure what to do next. They focus too narrowly and specifically on details of the case.")

G. ***The same cases discussed previously are presented to the expert's best student.*** We find which phrases have been picked up ("establish a data base") and how the student carries out the strategies he has learned. For example, a student might verbalize his reasoning more slowly and carefully, providing some details that the expert skips over.

H. ***We discuss each rule with the expert, grouping them according to the hypotheses they support*** (e.g., rules that conclude "bacterial meningitis"). From this analysis, we fill in the structure of data and the hypothesis space (e.g, we find out about different kinds of compromised hosts) and acquire a support story for each rule (why it is believed to be correct)[12]. By asking "when would you think about requesting this datum," we are able to cross check our strategic concepts and rules.

In summary, the methodology used to develop NEOMYCIN was task-oriented, namely to acquire the knowledge to place MYCIN's rules in order so they were more useful for teaching. We originally intended to simply "clean up" the rules, but decided that a more radical change in MYCIN's control structure was called for (use predominantly forward-directed reasoning instead of backward chaining).

To implement the expert's strategy, we had to translate his task statements ("establish the etiology") into more procedural terms ("establish a grouping of possibilities by confirming a path upwards in the hierarchy"). The idea that the initial problem formulation takes the expert into the middle of an etiological hierarchy was not stated by the expert. In fact, the concept of "initial problem formulation" came from previous work in problem-solving. [13]

The general methodology that we are following is summarized below. NEOMYCIN development is now iterating in steps 4 to 6.

1. Formulate design guidelines
   (This is the task orientation: What should the system do?
   Who will use it? This conception may change over time.)

2. Model system on paper (hand simulations) (steps 4-6)
   (This may take several months or more than a year,
   including the experiments described in this section.)

3. Code/modify program
   (including simplifications for elegance)

[12] We discovered that some rules were redundant or simply encoded incorrectly; some problem situations were not considered; some rules were "folklore" and not worth teaching.

[13] Significantly, the expert did tell students "you have to search the tree of possibilities," so he knew something about how he organized his knowledge.

4. Experiment with program
   -- *observe behavior on test cases*

5. Analyze program behavior (to determine shortcomings)
   -- *determine* appropriateness (expert perspective)
   -- *assign credit and blame* to code sections, determining if there is a programming error or shortcoming in the general theory or domain specific knowledge

6. Theorize/reformulate model (to eliminate shortcomings)
   -- restate theory principles and/or collect domain knowledge through reading and dialogue with expert
   -- use, modify, and develop programming technology

7. go to step 3

Testing NEOMYCIN will cover both its performance (*comparing* it to MYCIN) and use for teaching (incorporating it in GUIDON version 2). We expect that our experience with students using GUIDON2 will *enable us to* both *refine* the expert model and to construct, perhaps as a variation of NEOMYCIN, a *preliminary* model of novice diagnostic thinking.

7 Methodological Pitfalls

In the course of developing a program like NEOMYCIN, it is possible to lose the way temporarily. The pitfalls of an AI orientation to Cognitive Science include the problems of introspection, non-empiricism, and over-formalization.

7.1 Introspection and representation

In order to understand what the *expert was teaching us*, we drew diagrams of the hierarchies of data, hypotheses, and *rule* generalizations. Then, in trying to understand the *expert's strategies, we found ourselves remembering* these diagrams, so we were unable to separate our interpretation of the expert's behavior from our evolving representation of his knowledge. In particular, we came to realize that the structures we had drawn could account for the expert's reasoning in multiple ways, and we had been mistaken to think that we were capturing structures that were isomorphic to something that was "in his brain."

Some examples of this phenomenon might be useful. When the expert learns that the patient has a fever, he frequently will ask for details (severity, periodicity, etc.). This is modelled in NEOMYCIN as "process" questions that are directly associated with the concept of "fever." Yet, one could also say that the expert is thinking about a particular cause of fever, so asks about severity, for example, to see if the fever confirms his guess. This is in fact how Ann D. Rubin [Rubin, 1975] interpreted this kind of question, and it is consistent with her general model of hypothesis formation. However, we found no reason to postulate the intermediate steps of reasoning (setting up an hypothesis), even though the follow-up question is relevant because it is potentially useful.

The point is that in interpreting expert behavior we can easily crank through the reasoning processes and knowledge structures we have already formalized, producing system performance that matches the expert's but which does not simulate his reasoning steps (associations). The cause of this problem is that people's associations can be ad hoc, made efficient through rote, and are not restricted to the principled structures of subtype, causality, process, etc. that we postulate in a system like NEOMYCIN. This is the idea that knowledge can be "procedurally attached" and doesn't need to be stepped through in declarative form [Winograd, 1975]. (Anderson's program for modelling learning is based on proceduralization of this form [Anderson, et al., 1980].)

In NEOMYCIN, we have attempted to capture the "compiled associations" of the expert, while labeling them to record their principled basis. Thus "acute and chronic," process terms, are placed in what should be a strictly causal network (the etiological taxonomy, Figure 9). Similarly, the expert doesn't always clearly distinguish between the concepts of "subtype" and "cause," so a principled representation which does make this distinction must be interpreted by procedures which blur the difference.

Our investigation indicates that people form associations on any useful basis, and it is not trivial to find principled theories for the basis of these associations. For example, Pople is trying to account for how classificational and causal know!edge are combined. Pople's concept of "bridge concepts" provides a first order theory of how "trigger associations" evolve by combining the two kinds of associations through a form of transitive closure [Pople, 1980]. However, this model predicts far more trigger associations than expert behavior demonstrates. We will need to refine this theory by appealing to notions of complexity and usefulness of triggers.

Similarly, we can find "proceduralized associations" which have been learned by rote instead of the kind of composition that Anderson's model describes. For example, an expert considering fungal meningitis tends to ask about travel first; considering virus, he asks about absenteeism in the schools (for a child); considering TB he asks about crowded conditions and previous illnesses. We can explain these questions in terms of the principle "try to confirm the enabling step of a causal process first." Thus, in infectious disease diagnosis one first tries to establish exposure to the causative agent. But this is a rationalization, for neither we nor the expert learned what questions to ask in this way.

In conclusion, one pitfall of modelling using the AI-oriented approach we describe is the tendency to be satisfied with a consistent, coherent model (a knowledge representation and model of reasoning for diagnosis, learning, explanation, etc.) that produces the same behavior as the expert. Because we can learn by rote and we are able to compose factual knowledge with procedure, an expert's associations may be more complex, and not fit the formal elegance of the program. But relying only on introspection, and introspectively observing that we can reach the same results as the expert by reasoning like the program, we can be mislead into thinking we have modeled his reasoning. More precise experimentation is necessary if we hanker after psychological validity.

7.2 Empiricism and technology

In developing the first version of GUIDON, we were dangerously close to saying that because we could relate a student's partial solutions to MYCIN's rules, we had an explanation of his reasoning--as if just because a model could be constructed by a program, it was accurate. Similarly, it is easy to suppose that when a program is able to parse a user's English sentence (as in MYCIN's question/answer module), it has determined what the user is trying to say. One never considers that the next question could be a restatement or request for clarification--it is just the "next input." In a variant of the introspective pitfall, the programmer is now thinking like his model of the machine. Rather than thinking in terms of what he can do with his representation (what is suggested by the technology), the AI-oriented Cognitive scientist must be oriented to the phenomenon he is trying to emulate. Simulating the program in the problem-solving environment (Section 6) is a valuable approach.

The technological pitfall is exacerbated by those who never get their program working, so they don't get the hard shock of empirical test. In short, a program isn't a "functional model" [Pylyshyn, 1978] if it isn't functional.

7.3 Validity and elegance

As in the hard sciences and mathematics, it is important that a computational model be formally simple and elegant. However, programming provides special opportunities for reframing and reorganization which adds nothing to the theory being programmed, and tends to even obscure its implementation. On the other hand, a theory sometimes profits from reorganization of the code that implements it, in the same way a physicist can find formal clarity by manipulating his equations, looking for symmetry and the like.

One measure of improvement is the perspicuity of the code. If the new rules, frames, or whatever make it easier for a colleague to understand the theory (to see the theory in the code), the representation (and accompanying interpretation) has probably been improved. For example, a programmer may rerepresent a single rule with multiple steps in its action as a set of ordered rules with identical premises, producing what he takes to be a more elegant representation with only single steps in each rule. But this obscures the simple idea of a procedure being a block of steps. More effort is required to interpret the code to see the procedure within it, just as the problem-solver would need to exert more effort to carry out the procedure. Requiring a rule for each step of the procedure therefore violates our understanding of the theory we are implementing, so we say that the representation is not improved.

8 Areas for Collaboration

In this section we will list some research problems that have been suggested by our work. In doing this, we have two purposes: first, to demonstrate that a computational model like NEOMYCIN can suggest new areas for psychological research, and second, to encourage non-AI Cognitive Scientists to contribute methodological ass'stance for attacking these problems. The list of research problems follows:

1. *The structure of working memory.* Is the differential a simple list? A hierarchy? Does it include a stack of goals? For example, when refining an hypothesis, moving down a hierarchy, how is each child visited in turn? By a strategy that iteratively focusses on siblings, as in NEOMYCIN, or by a separate, "saved" list of waiting hypotheses to consider?

2. *Identifying lines of reasoning.* The expert stated a rule generalization (Figure 9) which might be used in multiple ways. One could think in terms of "differential broadening factors," leading to consideration of "compromised host risk factors" (data orientation). Or one could think in terms of "unusual causes," leading to consideration of "gram-negative organisms" (hypothesis orientation). Is it possible to say that the expert is following one line of reasoning and not the other? Could he in some sense be doing one thing that combines the goal and method, namely "trying to broaden the differential by considering compromised host risk factors"? Is it possible to get at the expert's line of reasoning without being misled by his rationalization? Or is it wrong to say that there is some explicit, conscious line of reasoning that we can discover?

3. *The effect of problem context.* Our expert supplied details to make the cases presented to him seem more realistic ("I'm at the patient's bedside" or "I'm in the emergency room and this patient comes up to me, accompanied by her mother"). Presenting a case twice, separated by many months, we saw that this story can change the expert's approach, even leading him to explore completely different hypotheses. How does the expert's imagination of the situation affect his reasoning? What variables must be specified to control for this effect?

4. *Clustering of hypotheses for manageability.* One diagnostic task is to refine a category by considering what causes it. Thus, the physician considers the types of chronic meningitis. However, a physician does not run through the several dozen organisms that might be causing bacterial meningitis. He thinks in terms of common and unusual causes to make the set more manageable. What happens when there are too many common causes to entertain? What other kinds of groupings are useful?

5. *Experimentally verifying diagnostic strategy.* How can we test NEOMYCIN's diagnostic strategy? For example, how do we confirm that focussing on an hypothesis and asking a question to confirm it are best described as two separate decisions, made independently? Or that an expert requests details before following up on the implications of data (process-oriented questions before making associations with hypotheses)? How can we test the control structure of strategies: a pure production system at the task level, tasks arranged hierarchically, and inherited "abort conditions"?

6. *Explanatory theory of strategy.* Can we construct a principled, explanatory theory that could in some sense generate the diagnostic meta-strategy? Viewing the processor ideas as constraints--a differential (working memory), focussed activation, hierarchical problem space and problem features, trigger associations, and strategic control--how do we derive a diagnostic procedure? For example, "Reviewing the differential" is not motivated by computational needs, but is a reflection of human forgetting. Rather than viewing this as a "forced imperfection" in the system, the review process (and indeed, the structure of the differential) might follow from a deeper model for retrieval of disease knowledge, along the lines of Lehnert's model of question answering [Lehnert, forthcoming].

7. *Modeling belief.* What makes an expert believe that a hypothesis is confirmed or

unlikely? Are there general principles for dealing with missing data, for knowing when to drop a losing line of inquiry, or to return to a previously discarded hypothesis?

8. ***Shifts of attention and noticing subproblems.*** When the problem solver gets more data, he may be receiving information that supports a hypothesis he is not currently considering. What determines whether he does/can shift attention temporarily? The NEOMYCIN model allows for focused associations to other hypothesis, but does not allow for "filing a reminder" to take something up later or noticing that a hypothesis is ruled out, so it is not considered later. What does the problem solver notice about other parts of the problem as he moves along and what kinds of notes does/can he make to himself to affect his performance later? What kinds of errors might shifts of attention cause? How does the problem solver avoid retracing his steps? If the current differential is poorly grouped, circumstantial evidence might support widely different hypotheses. Might this ambiguity be a likely point of error, in which one of the interpretations is missed? Are there metacognitive strategies for checking these errors?

9. ***Effect of level of abstraction on problem formulation.*** In discussing the same case separated by the period of months, the expert stated his initial differential (guess) differently. In one case he said "mass lesion." In the other case he broke this down into into subtypes. Very clearly, stating the subtypes brought other associations to mind, leading to a quite different exploration (using the same strategies). How can we account for this choice in level of abstraction? There is a clear trade-off, for the expert forgot to consider a traumatic problem when he was so busy reciting and considering the subtypes of mass lesion. What reasoning strategies do people use to maintain a manageable level of abstraction in working memory? What errors occur?

10. ***Observation strategies.*** We need to deal with the richness of the data collection procedure: partial stories are corrected later, making backtracking necessary; data must be verified; questions must be asked so they are understandable to the layman; therapeutic benefit, urgency, and availability of medical equipment must be factored in. Expertise surely requires a good deal of common sense. Just how the two are cross-related and build upon one another are difficult questions.

9 The Prospects for Collaboration

In carrying out the NEOMYCIN research, we have not had as many collaborative discussions as might have been useful. Few computer science graduate students, the most likely collaborators, have the necessary LISP programming experience, a background in AI techniques, a willingness to learn medical technology, and an inclination to do psychological research. Therefore, the most immediate methodological problem we face is superficial: a lack of trained people to share in the research. But what kind of collaboration is possible? Should we think of cognitive scientists as hybrids, or as specialists sharing in a common project?

Looking at the fields of cognitive psychology and AI today we find a wide spectrum of interests and methods, particularly along the dimensions of experimentation and programming. In cognitive psychology, we find, for example, Bower at one end, doing traditional psychology experiments and no programming, but making some use of AI concepts [Bower, 1981]. In the middle, we find someone like Feltovich, doing traditional experiments, but whose analyses and questions tend to be based in information processing terminology. At the other end, John Anderson is experimenting and writing programs, to the extent that people in computer science might think of him as being in AI.

In AI we find the same kind of spectrum. On the one hand, we find researchers with a psychological bent whose main goal is to build a working program, but who periodically say "It would be interesting to find out if people work this way" (e.g., [Fahlman, 1980] [Friedland, 1979]). This group includes the "knowledge engineers" [Feigenbaum, 1977], who have practical objectives, have fears about "listening to experts too closely" ("experts *can't really* explain how they reason"), and avoid the "paper modelling" of the psychologists. They want to build useful tools, therefore they are concerned with difficult, realistic problems (and never toy blocks). They want programs to be better than people, involving formalization of computational methods that perhaps people don't and can't use. Experimentation, to determine "what anybody's grandmother could have said," (as Gordon Bower puts it) is unnecessary. Talk to an expert and incorporate his heuristics. Test the program by asking the expert to point out shortcomings.

Finally, we find AI researchers using the behavioral studies of the cognitive psychologists to build a complex system for doing some real task (e.g., NEOMYCIN, [Lehnert, 1980]). These researchers are output-oriented like the first group, but their task involves human interaction in such a way that the program's reasoning should model human performance. This group also includes researchers who believe that the performance of AI programs can *be enhanced* if we better understand how people solve problems. When they listen to an expert, they are oriented to understanding how he is reasoning, not simply filling in their representation of slots, rules, etc. Potentially, this group could include any researcher in AI; work in learning, natural language understanding, and intelligent tutoring systems seems especially likely to benefit from cognitive studies.

In considering collaboration between AI and other fields in cognitive science, we should consider that people differ along these dimensions of interest and methodology. It is not at all clear that only people doing both experimentation and programming should be called cognitive scientists. It seems more likely that cognitive science will be made of people using interdisciplinary analogies and sharing research results.

The easiest form of collaboration is by evolution of common interests. We may not talk to each other directly very often, but we will communicate in the literature, translating ideas to our own application. For example, this is the way in which GUIDON research benefits from the work of Tversky concerning biases in human judgment [Tversky & Kahneman, 1977].

A second possibility is "mission-based" collaboration, in which we work together on a single project, sharing tasks according to our expertise. We might work in parallel--we might work with someone to precisely define a problem (such as those in Section 8) and months later he would return with experimental results.

It is important to remember the dialectic power of a program. The strength of cognitive science is surely in the way theories are changed and suggested by the very process of building computational models. Besides worrying that perhaps not enough formal experimentation is being done, we should be concerned that not enough cognitive scientists are writing programs, or helping to write programs. Too often experimental analysis seems to fall short by not being precise enough to be programmable. Or the simplifications to make an experiment tenable eliminate the very points that we need to build a working system (as fixed-order experiments in medical diagnosis eliminate focusing and data selection strategies).

Within the GUIDON/NEOMYCIN project, the experimentation that we do in the future, outside of continuing to interview experts, will consist of having students use GUIDON2. In many respects, these trials will resemble the experiments carried on by Feltovich, and others. (As his experiments have prepared us for the kinds of diagnostic errors students make.) Our theory of knowledge representation and strategies, and our lower-level concepts of the working memory and control structure will evolve as we change the program to meet the needs of the task. It is an open question just how detailed an "explanatory theory" is needed to build a reasonably effective intelligent tutoring system. In our collective work on diagnostic tasks, "bugs" and epistemology, we are already going beyond what the average teacher knows about reasoning. As the knowledge engineers, we reach for computational methods that surpass human expertise. However, in building an intelligent tutoring system, it is not sufficient to seek improvements in formal efficiency and elegance alone; we must also ask why people fail.

10 Glossary

An attempt is made here to generalize terminology beyond the medical application, though the reader should realize that some definitions are peculiar to our research project and others have a slightly different meaning in other areas of AI.

causal rules -- productions of the form, "If A then B" with the interpretation that "A is caused by B."

compiled association -- composition of a chain of productions into a single production, e.g., "If A then B" and "If B & C then D" might be compiled to "If A & C then D."

compromised host -- in medicine, a patient in a weakened condition that increases susceptibility to disease.

data -- facts about a problem in the form of direct measurements or circumstantial evidence.

differential -- a list of hypotheses that the problem solver is considering as possible solutions to the diagnostic problem.

diagnostic problem -- a situation, entity or event which the problem solver attempts to explain (characterize its nature) by observing its appearance and behavior over time.

disease -- in general, some underlying condition or process in a system which has an undesirable effect on the system.

disease process knowledge -- descriptive facts about diseases that have been previously observed in a system, along the lines of how the disease is caused and how it affects the system over time.

domain-dependent knowledge -- with respect to a given kind of diagnostic problem (e.g., electronic troubleshooting) and a given problem being diagnosed (e.g.., a Zenith computer terminal), those facts about the design of the system and its functionality, as well of scientific theories pertaining to its operation, that are useful for explaining how the system operates.

domain-independent knowledge -- facts and reasoning procedures brought to bear in problem solving which are not domain-dependent.

etiological taxonomy -- a hierarchy of diseases or possible causes of a diagnostic problem, in which the leaf nodes of the hierarchy are well-defined specific causes and intervening nodes are abstract categories of diseases.

expert -- a problem solver with sufficient knowledge to make correct diagnoses a high percentage of the time and to know when a problem cannot be confidently solved using the knowledge available to him.

expert system -- an AI computer program that is designed to solve problems at the expert level in some scientific, mathematical, or medical domain.

forward-directed inferences -- associations between data and hypotheses that are made by the problem solver at the time new data comes to his attention.

group and differentiate -- a diagnostic strategy which attempts to compact the differential so the hypotheses under consideration fall under a single node in the etiological taxonomy, generally by ruling out alternatives through discriminating data collection.

hypothesis -- a disease or more general causal category that the problem solver is considering as a solution of the diagnostic problem.

intelligent tutoring system -- an expert system whose domain of expertise is teaching, containing an expert system within it relevant to the area the tutoring system is teaching about.

interpreter -- a program that generally follows a simple control policy for applying knowledge to the problem at hand. The interpreter for disease knowledge determines how new problem data leads to inferences being made to augment working memory. The interpreter for strategical knowledge determines how planning knowledge is used for collecting new data or changing the phase of problem solving.

knowledge base -- domain-dependent knowledge represented in various AI formalisms.

knowledge engineering -- the art of building expert systems by working with experts to codify their knowledge.

meta-strategy -- a hierarchy of general tasks related by meta-rules, by which a problem solver directs his attention during diagnosis.

problem formulation -- the task of characterizing a diagnostic problem so that the correct etiological category is brought into the differential.

procedurally embedded -- knowledge that is implicit in the design of a program; for example, the rationale for ordering a sequence of steps in a particular way. A procedure is represented *declaratively* if the knowledge behind its design is explicitly represented in the system so that an interpreter can be applied to the design and domain knowledge to execute the procedure.

production rule -- an association of the form, "If A then B," whose interpretation is such that when A is considered, believed, or accomplished by the problem solver, it is valid (according to some unspecified justification) to consider, believe, or achieve B.

screening relation -- an association between data of the form, "A screens (for) B," with the interpretation that A should be considered before B with the justification that B might be derived from knowledge of A. For example, the sex of a patient screens for whether or not the patient is pregnant.

structural knowledge -- any organizational constructs based on domain-independent relations ("sibling of," "location," "process question follow-up," "screening question"), used by a meta-strategy to index domain-dependent knowledge.

subgoal -- in MYCIN, a reasoning step that appears as a clause in the premise of some production rule; for example in the rule "if A & B then C," A and B are subgoals.

subtype -- a relation between disease categories, synonymous with "kind of."

top-down refinement -- the diagnostic strategy of searching the etiological taxonomy in breadth-first manner starting at some node of the tree; called "refinement" because each level of the tree specifies a finer or more precise diagnosis.

triggers -- production rules of the form, "if A then B" where A is a conjunction mentioning problem data which are said to "trigger" or "suggest directly" the hypothesis B, which appears in the etiological taxonomy.

References

[Anderson, at al., 1980] Anderson, J. R., Greeno, J. G., Kline, P. J., & Neves, D. M. Acquisition of problem-solving skill. Carnegie Mellon Technical Report 80-5. To appear in J. R. Anderson (Ed.), *Cognitive Skills and their Acquisition.*

[Boden, 1977] Boden, M. A. *Artificial Intelligence and Natural Man.* New York: Basic Books, 1977.

[Bower, 1981] Bower, G. H. Mood and Memory. *American Psychologist*, 1981, 36(2) 129-148.

[Brown, Burton, & Bell, 1974b] Brown, J. S., Burton, R. R., & Bell, A. G. *Sophie: A Sophisticated Instructional Environment for teaching electronic troubleshooting (An example of AI in CAI)* (BBN Report No. 2790). 1974.

[Brown, Collins, & Harris, 1977] Brown, J. S., Collins, A., & Harris, G. Artificial intelligence and learning strategies. In O'Neill (Ed.), *Learning Strategies.* New York: Academic Press, 1977.

[Brown & Burton, 1978] Brown, J. S. & Burton, R. R. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 1978, 2(2), 155-192.

[Brown & VanLehn, forthcoming] Brown, J. S. & VanLehn, K. Repair theory: a generative theory of bugs in procedural skills. To appear in *Cognitive Science.*

[Brown & Sleeman, 1981] Brown, J. S. & Sleeman, D. *Intelligent Tutoring Systems.* London: Academic Press, forthcoming, 1981.

[Burton, 1979] Burton, R. R. An investigation of computer coaching for informal learning activities. *The International Journal of Man-Machine Studies*, 1979, *11*, 5-24.

[Clancey, 1979a] Clancey, W. J. Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies*, 1979, *11*, 25-49.

[Clancey, 1979b] Clancey, W. J. Transfer of Rule-Based Expertise through a Tutorial Dialogue. Computer Science Doctoral Dissertation, Stanford University, STAN-CS-769, August, 1979.

[Clancey and Letsinger, 1981] Clancey, W. J. and Letsinger, R. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. *Proceedings of the Seventh IJCAI*, 1981, 829-836.

[Clancey, forthcoming] Clancey, W. J. The epistemology of a rule-based expert system. Submitted in the *Journal of Artificial Intelligence.*

[Collins, 1976] Collins, A. Processes in acquiring knowledge. In R. C. Anderson, R. J. Spiro, & W. E. Montague (Eds.) *Schooling and the acquisition of knowledge*. Hillsdale, NJ: Erlbaum Assoc., 1976, pp. 339-363.

[Davis, 1979] Davis, R. *Interactive transfer of expertise: acquisition of new inference rules. Journal of Artificial Intelligence*, 1979, *12*, 121-157.

[Davis, 1976] Davis, R. *Applications of meta-level knowledge to the construction, maintenance and use of large knowledge bases* (STAN-CS-76-552, HPP-76-7). Stanford University, July 1976.

[de Kleer, 1979] de Kleer, J. The *origin and resolution of* ambiguities in causal arguments. *Proceedings of the Sixth IJCAI*, 1979, 197-203.

[Elstein, Shulman, & Sprafka, 1978] Elstein, A. S., Shulman, L. S., & Sprafka, S. A. *Medical problem-solving: An analysis of clinical reasoning*. Cambridge: Harvard University Press, 1978.

[Fahlman, 1980] Fahlman, S. E. Design Sketch for a million-element NETL machine. *Proceedings of the First Annual National Conference on Artificial Intelligence*, 1980, 249-252.

[Feigenbaum, 1977] Feigenbaum, E. A. The art of Artificial Intelligence: I. Themes and case studies of knowledge engineering. *Proceedings of the Fifth IJCAI*, 1977, 1014-1029.

[Feltovich, et al., 1980] Feltovich, P. J., Johnson, P. E., Moller, J. H., & Swanson, D. B. *The role and development of medical knowledge in diagnostic expertise*. Presented at the 1980 Annual meeting of the Americal Educational Research Association.

[Friedland, 1979] Friedland, P. Knowledge-based experiment design in molecular genetics. *Proceedings of the Sixth IJCAI*, 1979, 285-287.

[Greeno, 1976] Greeno, James G. Cognitive objectives of instruction: Theory of knowledge for solving problems and answering questions. In Klahr (Ed.), *Cognition and Instruction*. Hillsdale, NJ: Erlbaum Associates.

[Greeno & Brown, forthcoming] Greeno, J. G. and Brown, J. S. Theories of Competence. *This volume*.

[Kintsch, et al., forthcoming] Kintsch, W., Miller, J., and Polson, P. Problems of Methodology in Cognitive Science. *This volume*.

[Kosslyn, 1980] Kosslyn, S. M. *Image and Mind*. Cambridge: Harvard University Press, 1980.

[Lehnert, 1980] Lehnert, W. Narrative text summarization. *Proceedings of the First Annual National Conference on Artificial Intelligence*, 1980, 337-339.

[Lehnert, forthcoming] Lehnert, W. Paradigmatic Issues in Cognitive Science. *This volume.*

[McDermott, 1980] McDermott, J. R1: A rule-based configurer of computer systems. Department of Computer Science, Carnegie-Mellon University, CMU-CS-80-119, April, 1980.

[Miller, 1975] Miller, P. B. *Strategy selection in medical diagnosis.* Project MAC, Massachusetts Institute of Technology, MAC TR-153, September 1975.

[Nii & Feigenbaum, 1978] Nii, H. P. & Feigenbaum, E. A. Rule-based understanding of signals. In Waterman and Hayes-Roth (Eds.) *Pattern-directed Inference Systems*, New York: Academic Press, 1978.

[Norman & Rumelhart, 1975] Norman D. A. & Rumelhart, D. E. *Explorations in Cognition.* San Francisco: Freeman, 1975.

[Pauker and Szolovits, 1977] Pauker, S. G. & Szolovits, P. Analyzing and simulating taking the history of the present illness: Context Formation. In Schneider/Segvall Hein (Eds.), *Computational Linguistics in Medicine.* North-Holland Publishing Company, 1977, 109-118.

[Pople, 1980] Pople, H. E. *Heuristic methods for imposing structure on ill-structured problems: the structuring of medical diagnostics.* To appear in a monograph by the American Assoc. for Adv. of Science.

[Pylyshyn, 1978] Pylyshyn, Z. W. Computational models and empirical constraints. *The Behavioral and Brain Sciences*, 1978, *1*, 93-127.

[Rubin, 1975] Rubin A. D. *Hypothesis formation a evaluation in medical diagnosis.* Artificial Intelligence Laboratory, MIT, Technical Report AI-TR-316, January 1975.

[Rumelhart & Norman, 1980] Rumelhart, D. E. & Norman, D. A. *Analogical processes in learning.* University of California, San Diego, Technical report 8005, September 1980.

[Scott, et al., 1977] Scott, A. C., Clancey, W. J., Davis, R., Shortliffe, E. H. Explanation capabilities of production-based consultation systems. *American Journal of Computational Linguistics*, 1977, Microfiche 62.

[Shortliffe, 1976] Shortliffe, E. H. *Computer-based medical consultations: MYCIN.* New York: Elsevier, 1976.)

[Stefik, 1979] Stefik, M. An examination of a frame-structured representation system. *Proceedings of the Sixth IJCAI*, 1979, 845-852.

[Stevens, Collins, & Goldin, 1978] Stevens, A. L., Collins, A., & Goldin, S. *Diagnosing student's misconceptions in causal models* (BBN Report No. 3786). 1978.

[Swanson, et al., 1977] Swanson, D. B., Feltovich, P. J., & Johnson, P. E. Psychological analysis of physician expertise: implications for design of decision support systems. *MEDINFO77*, 1977, 161-164.

[Tversky & Kahneman, 1977] Tversky A. & Kahneman, D. Judgment under uncertainty: heuristics and biases. In P. N. Johnson-Laird and P. C. Wason (Eds.) *Thinking: Readings in Cognitive Science*, Cambridge University Press, 1977.

[van Melle, 1980] van Melle, W. A domain-independent production-rule system for consultation programs. Computer Science Doctoral Dissertation, Stanford University, in press, 1980.

[Wescourt, Beard, & Gould, 1977] Wescourt, K. T., Beard, M., & Gould, M. *Knowledge-based adaptive curriculum sequences for CAI: Application of a network representation* (Tech. Rep. 288). Institute for Mathematical Studies in the Social Sciences, Stanford University, 1977.

[Winograd, 1975] Winograd, T. Frame Representations and the Declarative/Procedural Controversy. In D. G. Bobrow & A. Collins (Eds.), *Representation and Understanding*. New York: Academic Press, 1975, 185-210.

[Yu, et al., 1979a] Yu, V. L., Buchanan, B. G., Shortliffe, E. H., Wraith, S. M., Davis, R., Scott, A. C., & Cohen, S. N. Evaluating the performance of a computer-based consultant. *Computer Programs in Biomedicine*, 1979, 9, 95-102. (a)

[Yu, et al., 1979b] Yu, V. L., Fagan, L. M., Wraith, S. M., Clancey, W. J., Scott, A. C., Hannigan, J. F., Blum, R. L., Buchanan, B. G., Cohen, S. N. Antimicrobial selection by a computer--a blinded evaluation by infectious disease experts. *Journal of the American Medical Association*, 1979, *242*, 1279-1282. (b)