Report 83-18
Stanford -- KSL

Abstract Explanations of Strategy in a Diagnostic Consultation System.
Diane W. Hasling,
Apr 1983

HPP-83-18

# ABSTRACT EXPLANATIONS OF STRATEGY IN A DIAGNOSTIC CONSULTATION SYSTEM

**Diane Warner Hasling**

Heuristic Programming Project
Computer Science Department
Stanford University
Stanford, CA 94305

## ABSTRACT

This paper presents the explanation system for NEOMYCIN* , a medical consultation program. A consultation program plays the role of an expert to assist a user in solving a problem. An explanation of strategy describes the plan the program is using to reach a solution. Such an explanation is usually concrete, referring to aspects of the current problem situation. Abstract explanations articulate a general principle, which can be applied in different situations; such explanations are useful in teaching and in explaining by analogy. We describe the aspects of NEOMYCIN that make abstract strategic explanations possible--the representation of strategic knowledge explicitly and separately from domain knowledge--and demonstrate how this representation can be used to generate explanations.

## I INTRODUCTION

The ability to explain reasoning and actions is usually considered an important component of any expert system. An explanation facility is useful on several levels: it can help knowledge engineers to debug and test the system during development, assure the sophisticated user that the system's knowledge and reasoning process is appropriate, and instruct the naive user or student about the knowledge in the system. Several approaches have been used in existing explanation systems. For example, Shortliffe (Shortliffe, 1976) and Davis (Davis, 1976) introduced the idea of generating explanations by translating rules that direct a consultation. Swartout (Swartout, 1981) uses an automatic programming approach to create a static "refinement structure", which can be examined during the consultation to provide justifications of the compiled code.

A *strategy* is "a careful plan or method, especially for achieving a end." To *explain* is "to make clear or plain; to give the reason for or cause of."** Thus in a *strategic explanation* we are trying to make clear the plans and methods used in reaching a goal, in NEOMYCIN's case, the diagnosis of a medical problem. One could imagine explaining an action in at least two ways. In the first, the specifics of the situation are cited, with the strategy remaining relatively implicit. For example, "I'm asking whether the patient is receiving any medications in order to determine if she's receiving penicillin." In the second approach, the underlying strategy is made explicit; "I'm asking whether the patient is receiving any medications because I'm interested in determining whether she's receiving penicillin. *I ask a general question before a specific one when possible.*" This latter example is the kind of strategic explanation we want to generate. The general approach to solving the problem is mentioned, as well as the action taken in a particular situation. Explanations of this type allow the listener to see the larger problem-solving approach and thus to examine, and perhaps learn, the strategy being employed.

Our work is based on the hypothesis that an 'understander' must have an idea of the problem-solving process, as well as domain knowledge, in order to understand the solution or solve the problem himself (Brown, 1978). Specifically, research in medical education (Elstein, 1978), (Benbassat, 1976) suggests that we state heuristics for students, teaching them explicitly how to acquire data and form diagnostic hypotheses. Other AI programs have illustrated the importance of strategies in explanations. SHRDLU (Winograd, 1972) is an early program that incorporates history keeping to provide WHY/HOW explanations of procedures used by a 'robot' in a simulated BLOCKSWORLD environment. The procedures of this robot are specific to the environment; consequently, abstract explanations such as "I moved the red block *to achieve preconditions of a higher goal*" are not possible. CENTAUR (Aikins, 1980), another medical consultation system, explains its actions in terms of domain-specific operations and diagnostic prototypes. Swartout's XPLAIN program refers to domain principles--general rules and constraints about the domain--in its explanations. In each of these programs, abstract principles have been instantiated and represented in problem-specific terms.

NEOMYCIN generates strategic explanations from an *abstract* representation of strategy. In contrast with other approaches, this strategic knowledge is completely separate from the domain knowledge. This general strategy is instantiated dynamically as the consultation runs. Thus when the program discusses the problem solution, it is able to state a general approach, as well as how it applies in concrete terms.

## II HOW STRATEGIC EXPLANATIONS ARE POSSIBLE -- THE NEOMYCIN SYSTEM

MYCIN (Shortliffe, 1976), the precursor of NEOMYCIN, is unable to explain its strategy because much of the strategic information is implicit in the ordering of rule clauses (Clancey, 1983a). In NEOMYCIN, the problem-solving strategy is both explicit and general. This section provides an overview of the representation of this strategy in NEOMYCIN, since this is the basis for our strategic explanations. Other aspects of the system, such as the disease taxonomy and other structuring of the domain knowledge, are described in (Clancey, 1981).

NEOMYCIN's strategy is structured in terms of *tasks*, which correspond to metalevel goals and subgoals, and metalevel rules (*metarules*), which are the methods for achieving these goals. The metarules invoke other tasks, ultimately invoking the base-level interpreter to pursue domain goals or apply domain rules. Figure 1 illustrates a portion of the task structure, with metarules linking the tasks. The entire structure currently includes 30 tasks and 74 metarules.

---

*The design and implementation of the NEOMYCIN explanation system has been supervised by William J Clancey. We gratefully acknowledge the assistance of Bruce Buchanan, Glenn Reinech, Ted Shortliffe, and Derek Sleeman. This research has been supported in part by ONR and ARI contract N00014-79C-0302. Computational resources have been provided by the SUMEX-AIM facility (NIH grant RR00785).

**Webster's New Collegiate Dictionary

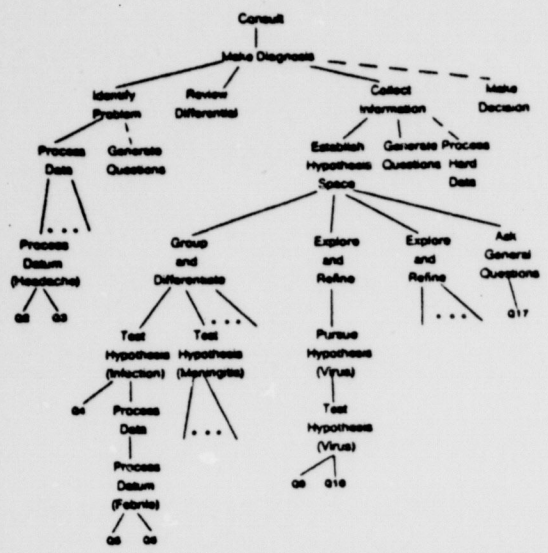


**Figure 1: Invocation of tasks in the example NEOMYCIN consultation**
Question numbers correspond to questions asked in the consultation, solid lines show tasks actually done, dotted lines those which might be done. Note how tasks such as TEST HYPOTHESIS are invoked multiple times by a given task as well as by different tasks.

This task structure represents a general diagnostic problem-solving method. Although our base-level for development has been medicine, none of the tasks or metarules mention the medical domain. As a result the strategy might be ported to other domains. (See (Clancey, 1983b) for further discussion.)

An ordered collection of metarules constitutes a procedure for achieving a task. Each metarule has a premise, which indicates when the metarule is applicable, and an action, indicating what should be done whenever the premise is satisfied. Figure 2 is a high-level abstraction of a task and its metarules. The premise looks in the domain knowledge base or the problem-solving history for findings and hypotheses with certain properties, for example, possible follow-up questions for a recent finding or a subtype of an active hypothesis. Associated actions would be to ask the user a question or call a task to refine the hypothesis under consideration.

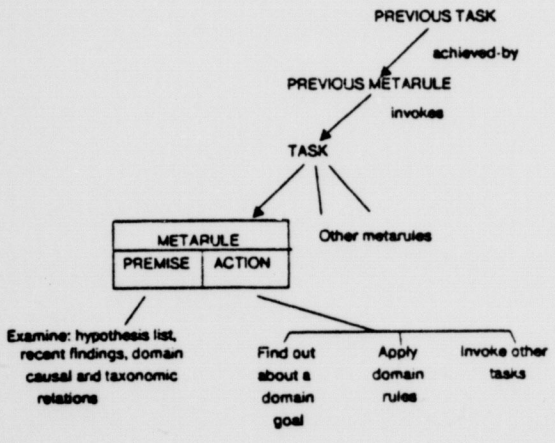


**Figure 2: Abstraction of a Task and Its Metarules**

Testing an hypothesis is just one reason for asking a question. Others are:

- Initial identification of the problem
- Follow-up questions to specify recent information in greater detail
- Questions triggered by recent data that suggested an hypothesis
- General questions to determine the completeness of the case history
- Screening questions to determine the relevance of desired data
- Questions asked while pursuing subgoals of the domain.

The number of reasons for asking questions testifies to the complexity of NEOMYCIN's diagnostic strategy and illustrates why strategic explanations are both useful and interesting.

## III NEOMYCIN'S STRATEGIC EXPLANATIONS

### A. Design Criteria

In determining what the program should explain and how it should be explained, we used several design criteria:

- Explanations should *not presuppose any particular user population.* The long-range goal of this research is to use NEOMYCIN as the foundation of a teaching system. At that point the strategic explanations developed here will be used to teach the strategy to students to whom it might be unfamiliar. Techniques used to generate explanations should be flexible enough to accommodate a model of the user.
- Explanations should be *informative*; rule numbers or task names are not sufficient.
- Explanations should be *concrete or abstract*, depending upon the situation. Thus it must be possible to produce explanations in either form. This should facilitate understanding both of the strategy and how it is actually applied.
- Explanations should be *useful for the designer, as well as the end user* of NEOMYCIN. The vocabularies of computer science and an application domain, such as medicine, are different in many ways. People tend to be most comfortable with the vocabulary of their field; the system should have the flexibility to accommodate a user-dependent choice of terminology.
- Explanations should be *possible at the lowest level of interest*; the "grain level" should be fine enough to permit this. To allow for use in debugging, we chose the level of rules and tasks as our lowest level. Higher level explanations can later be generated by omitting details below the appropriate level.

### B. WHY and HOW Explanations

Excerpts from a NEOMYCIN consultation illustrate the strategic explanation capability most clearly. User input follows the double asterisks (**) and is shown in capital letters. The consultation begins:

```
Please enter information about the patient.

     Name                    Age        Sex        Race

1) ** MARY                   42         F          LATINO

Please describe the chief complaints:

** HEADACHE
** STIFF-NECK
** NAUSEA
**

2) How many days has Mary's headache lasted?
** 10

3) How severe is Mary's headache (on a scale of 0 to 4
   with 0 for very mild and 4 for very severe)?
** WHY

[i.e. WHY is it important to determine the severity of
     Mary's headache?]

[1.0] We are trying to characterize the data just
     obtained and develop the differential.

Headache severity is a characterizing feature of headache.

[back to question 3...]
```

To explain WHY something is being done by NEOMYCIN, we first restate the question as, "WHY are we trying to do this task?" Because each task is invoked by a metarule in another task (see Figure 2), we answer the query by stating the task and metarule that invoked the task in question. At the lowest level, there are implicit tasks of requesting data and applying rules. The template for an initial WHY explanation is:

```
<request for data>
** WHY

[i.e. WHY is it important to determine <data>?]

[1.0] <immediately preceding task>

<what is true about the domain knowledge base or the
problem-solving history that enables the metarule that
accomplishes this task to succeed>
```

In this case the current metarule is:

```
METARULE073
IF there is a datum that can be requested that is a
     characterizing feature of the recent finding
     that is currently being considered,
THEN find out about the datum.
```

We mentioned earlier the idea of providing concrete or abstract explanations as appropriate. In the context of a diagnostic consultation, it seems reasonable to provide concrete explanations when possible. Abstract explanations can always be produced, however. The abstract explanation in this situation is:

```
**ABSTRACT

[i.e. WHY is it important to determine the severity of
     Mary's headache?]

[1.0] We are trying to characterize the data just obtained
     and develop the differential.

A datum that can be requested is a characterizing feature
     of the recent finding that is currently being
     considered.
```

When subsequent WHY's are given, we ascend the strategy tree, as illustrated below.

```
** WHY

[i.e. WHY are we trying characterize the data just
     obtained and develop the differential?]

[2.0] We are trying to give thorough consideration
     to any recent findings.

The fact that Mary has suffered from headaches recently
     is a new finding, which first must be characterized
     and then used to assist the diagnosis.

[back to question 3...]

** WHY

[i.e. WHY are we trying give thorough consideration to
     any recent findings?]

[3.0] We are trying to gather identifying information--
     the chief complaint and major problems.

The user must state the problem.

[back to question 3...]

** 3

DIFFERENTIAL:
(VIRUS 564) (CHRONIC-MENINGITIS 422)
```

Questions 4 and 14 below are both asked as a result of the TEST-HYPOTHESIS task. Notice how the explanations of this single task reflect the different situations of the two WHY questions. Different metarules apply--question 4 is based on a strong "triggering" relation, while question 14 is based on only moderate evidence.

```
4) Does Mary have a fever?
** WHY

[i.e. WHY is it important to determine whether Mary has
     a fever?]

[4.0] We are trying to decide whether Mary has an
     infection.

Fever is strongly associated with infection.

[back to question 4...]

** YES
   .
   .
   .

14) Does Mary have a history of granulomas on biopsy of
     the liver, lymph node, or other organ?
** WHY

[i.e. WHY is it important to determine whether Mary has a
     history of granulomas on biopsy of the liver, lymph
     node, or other organ?]

[19.0] We are trying to decide whether Mary has
     tuberculosis.

A history of granulomas on biopsy of the liver, lymph node,
     or other organ makes tuberculosis more likely.

[back to question 14...]
```

HOW is asked in terms of a previously mentioned task. The explanation presents any metarules associated with the relevant instance of the task that have been completed, as well as the one currently being executed. As a result of the structure of tasks and metarules, the WHY/HOW explanations have an interesting property. A WHY explanation essentially states the premise of the metarule; this is exactly the reason the metarule succeeded. A HOW explanation is a statement of the action of a metarule; this is exactly what was done.

In the next example, we are asking HOW about a low-level task. Note that the answer is at the lowest level of applying rules.

```
** HOW 19.0

[i.e. HOW are we trying to decide whether Mary has
     tuberculosis?]

We did the following:

[20.1] Applied a rule mentioning a causal precursor to
     tuberculosis (tb risk factor; rule525), and
[20.2] Gave thorough consideration to any recent findings.

[20.3] Applied a rule mentioning general evidence for
     tuberculosis (focal neurological signs; rule366), and
[20.4] Gave thorough consideration to any recent findings.

[20.6] Applied a rule mentioning general evidence for
     tuberculosis (compromised host; rule309), and
[20.6] Gave thorough consideration to any recent findings.

We are now:

[20.7] Applying a rule mentioning general evidence for
     tuberculosis (a history of granulomas on biopsy of
     the liver, lymph node, or other organ; rule002).

We will next:

[20.8] Give thorough consideration to any recent findings.

[back to question 14...]

** NO
```

The HOW question below refers to a high level task (ESTABLISH-HYPOTHESIS-SPACE) mentioned in the omitted sequence of WHY's. Notice that the explanation is at a high level and is entirely abstract.

```
17) Has Mary been hospitalized recently?
** WHY
   .
   .
   .

** HOW 26.0

[i.e. HOW are we trying to develop the differential using
     basic history and pertinent physical exam?]

We did the following:

[26.1] Got a general idea of the problem: categorized it
     into one of several pathogenic classes or disease
     loci, or both.

[26.2] Confirmed and refined the differential diagnosis
     through specific questions.

We are now:

[26.3] Rounding out the diagnostic information by looking
     generally into past medical history and by reviewing
     systems.

[back to question 17...]

** NO
```

NEOMYCIN uses an explanation approach similar to MYCIN's, that of explaining its actions in terms of goals and rules, so a brief comparison of the two systems is useful (Figure 3).
The structure of explanations is parallel, except that in MYCIN rules invoke subgoals through their premises, while NEOMYCIN metarules invoke subtasks through their actions. What makes NEOMYCIN's explanations different is that they are generated at the level of general strategies, instantiated with domain knowledge, when possible, to make them concrete.

| MYCIN | NEOMYCIN |
|---|---|
| Basic reasoning: goal -> rule -> subgoal | Basic reasoning: task -> metarule -> subtask |
| A goal is pursued to satisfy the premise of a domain rule (backward chaining) | A task is pursued when executing the action of a metarule (forward reasoning with rule sets) |
| To explain *why* a goal is pursued, cite the domain rule that uses it as a subgoal (premise) | To explain *why* a task is done, cite the metarule that invokes it (action) |
| To explain *how* a goal is determined, cite the rules that conclude it | To explain *how* a task is accomplished, cite the metarules that achieve it |

Figure 3: Comparison of MYCIN and NEOMYCIN Explanations

Besides these strategic WHY's and HOW's, the user can ask about the current hypothesis, the set of hypotheses currently being considered, and evidence for hypotheses at the domain level.

## C. Implementation Issues

We mentioned earlier that NEOMYCIN was designed with the intent of guiding a consultation with a general diagnostic strategy. A given task and associated metarules may be applied several times in different contexts in the course of the consultation, for example, testing several hypotheses. To produce concrete explanations, we keep records whenever a task is called or a metarule succeeds; this is sometimes called an *audit trail*. Data such as the focus of the task (e.g., the hypothesis being tested) and the metarule that called it are saved for tasks. Metarules that succeed are linked with any additional variables they manipulate, as well as any information that was obtained as an immediate result of their execution, such as questions that were asked and their answers. When an explanation of any of these is requested, the general translations are instantiated with this historical information.

Figure 4 presents several metarules for the TEST-HYPOTHESIS task translated abstractly.

```
METARULE411
IF The datum is question is strongly associated with the
     current focus
THEN Apply the related list of rules
Trans: ((VAR ASKINGPARM)(DOMAINWORD "triggers")(VAR CURFOCUS))

METARULE506
IF The datum in question makes the current focus more likely
THEN Apply the related list of rules
Trans: ((VAR ASKINGPARM) "makes" (VAR CURFOCUS) "more likely")
```

Figure 4: Sample NEOMYCIN Metarules for the TEST-HYPOTHESIS task

A sample of the audit trail created in the course of a consultation is shown in Figure 5; this is a snapshot of the TEST-HYPOTHESIS task after question 14 in the consultation excerpt. An example of how the general translations thus relate to the context of the consultation can be seen in the differing explanations for questions 4 and 14, both asked because an hypothesis was being tested.

In order to generate explanations using an appropriate vocabulary for the user, we've identified general words and phrases used in the translations that have parallels in the vocabulary of the domain. At the start of a consultation, the user identifies himself as either a "domain" or "system" expert. Whenever a marked phrase is encountered while explaining the strategy, the corresponding domain phrase is substituted for the medical expert. For example, "triggers" is replaced by "is strongly associated with" for the domain expert.

TEST-HYPOTHESIS

*STATIC PROPERTIES*

```
TRANS: ((VERB decide) whether * has (VAR CURFOCUS))
TASK-TYPE : ITERATIVE
TASKGOAL : EXPLORED
FOCUS : CURFOCUS
LOCALVARS : (RULELST)
CALLED-BY : (METARULE393 METARULE400 METARULE171)
TASK-PARENTS : (GROUP-AND-DIFFERENTIATE PURSUE-HYPOTHESIS)
TASK-CHILDREN : (PROCESS-DATA)
ACHIEVED-BY : (METARULE411 METARULE586 METARULE603)
DO-AFTER : (METARULE332)
```

*AUDIT TRAIL*

```
FOCUS-PARM : (INFECTIOUS-PROCESS MENINGITIS VIRUS
              CHRONIC-MENINGITIS MYCOBACTERIUM-TB)
CALLER : (METARULE393 METARULE400 METARULE171 METARULE171
              METARULE171)
HISTORY : [(METARULE411 ((RULELST RULE423)
                         (QUES 4 FEBRILE PATIENT-1 RULE423)))
           (METARULE411 ((RULELST RULE060)
                         (QUES 7 CONVULSIONS PATIENT-1
                           RULE060)))
           .
           .
           .
           (METARULE586 ((RULELST RULE525)
                         (QUES 11 TBRISK PATIENT-1 RULE525))
                    METARULE603
                    ((RULELST RULE366)
                     (QUES 12 FOCALSIGNS PATIENT-1 RULE366))
                    METARULE603
                    ((RULELST RULE309)
                     (QUES 13 COMPROMISED PATIENT-1 RULE309))
                    METARULE603
                    ((RULELST RULE002)
                     (QUES 14 GRANULOMA-HI PATIENT-1 RULE002]
```

Figure 5: Sample Task Properties

## IV LESSONS AND FUTURE WORK

The implementation of NEOMYCIN's explanation system has shown us several things. We've found that for a program to articulate general principles, strategy should be represented explicitly and abstractly. They are made explicit by means of a representation in which the control knowledge is explicit, that is, not embedded or implicit in the domain knowledge, such as in rule clause ordering. In NEOMYCIN this is done by using metarules, an approach first suggested by Davis (Davis, 1976). The strategies are made abstract by making metarules and tasks domain-independent. We've seen that it is possible to direct a consultation using this general problem-solving approach and that resulting explanations are, in fact, able to convey this strategy. As far as the utility of explanations of strategy, trials show that, as one might expect, an understanding of domain level concepts is an important prerequisite to appreciating strategic explanations.

In regard to representation issues, we've found that if control is to be assumed by the tasks and metarules, *all* control must be encoded in this way. Implicit actions in functions or hidden chaining in domain level rules lead to situations which do not fit into the overall task structure and cannot be adequately explained. This discovery recently encouraged us to implement two low-level functions as tasks and metarules, namely MYCIN's functions for acquiring new data and for applying rules. Not only do the resulting explanations reflect more accurately the actual activities of the system, they're also able to convey the purpose behind these actions more clearly.

There is still much that can be done with NEOMYCIN's strategic explanations. We mentioned that our current level of detail includes every task and metarule. We'd like to develop discourse rules for determining a reasonable level of detail for a given user. We also plan to experiment with summarization, identifying the key aspects of a segment of a consultation or the entire session. We might also explain why a metarule failed, why metarules are ordered in a particular way, and the justifications for the metarules. An advantage of our abstract representation of the problem-solving structure is that when the same procedure is applied in different situations, the system is able to recognize this fact. This gives us the capability to produce explanations by analogy, another area for future research.

## References


Aikins, J. S. *Prototypes and Production Rules: A Knowledge Representation for Computer Consultations.* PhD thesis, Stanford University, 1980. (STAN-CS-80-814).

Benbassat, J. and Schiffmann, A. An Approach to Teaching the Introduction to Clinical Medicine. *Annuals of Internal Medicine*, April 1976, *84(4)*, 477-481.

Brown, J. S., Collins, A., Harris, G. Artificial Intelligence and Learning Strategies. In O'Neil (editor), *Learning Strategies* . Academic Press, New York, 1978.

Clancey, W.J. and Letsinger, R. *NEOMYCIN: Reconfiguring a Rule-based Expert System for Application to Teaching*, in *Proceedings of the Seventh IJCAI*, pages 829-836, 1981.

Clancey, W.J. The Epistemology of a Rule-based Expert System: a Framework for Explanation. *Artificial Intelligence*, 1983, *20(3)*, 215-251.

Clancey, W.J. *The Advantages of Abstract Control Knowledge in Expert System Design.* To appear in *Proceedings of AAAI-83.*

Davis, R. *Applications of Meta-level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases.* PhD thesis, Stanford University, July, 1976. (STAN-CS-76-552, HPP-76-7).

Elstein, A. S., Shulman, L.S., and Sprafka, S.A. *Medical Problem Solving: An Analysis of Clinical Reasoning.* Cambridge, Massachusetts: Harvard University Press 1978.

Shortliffe, E.H. *Computer-based Medical Consultations: MYCIN.* New York: Elsevier 1976.

Swartout, W. R. *Explaining and Justifying Expert Consulting Programs*, in *Proceedings of the Seventh IJCAI*, pages 815-822, 1981.

Winograd, T. *Understanding Natural Language.* New York: Academic Press 1972.