6

Details of the Revised Therapy Algorithm

William J. Clancey

A program that is designed to provide sophisticated expert advice must cope with the needs of naive users who may find the advice puzzling or difficult to accept. This chapter describes additions to MYCIN that provide for explanations of its therapy decisions, the lack of which was a shortcoming of the original therapy recommendation code described in Section 5.4 of Chapter 5. It deals with an optimization problem that seeks to provide "coverage" for organisms while minimizing the number of drugs prescribed. There are many factors to consider, such as prior therapies and drug sensitivities, and a person often finds it hard to juggle all of the constraints at once. When the optimal solution is provided by a computer program, its correctness may not be immediately obvious to the user. This motivates our desire to provide an explanation capability to justify the program's results.

The explanation capability derives from two basic programming considerations. First, we have used heuristics that capture what expert physicians consider to be good medical practice. Thus, while the program is not designed to mimic the step-by-step problem-solving behavior of a physician, its chief decision criteria have been provided by expert physicians. It is accordingly plausible that the criteria will make sense to other physicians.

The second consideration is that the program must maintain records of decisions that were made. These are used for explaining what occurred

This chapter is an expanded version of a paper originally appearing in *Proceedings of the IJCAI 1977*. Used by permission of International Joint Conferences on Artificial Intelligence, Inc.; copies of the *Proceedings* are available from William Kaufmann, Inc., 95 First Street, Los Altos, CA 94022.

during the optimization process and why the output was not different. While the maintenance of records for explanation purposes is not new (e.g., see Winograd, 1972; Bobrow and Brown, 1975; Scragg, 1975a; 1975b), the means that we use to retrieve them are novel, namely a state transition representation of the algorithm. Our work demonstrates that a cleanly structured algorithm can provide both sophisticated performance and a simple, useful explanation capability.

6.1 The Problem

The main problem of the therapy selector is to prescribe the best drug for each organism thought to be a likely cause of the infection, while minimizing the total number of drugs. These two constraints often conflict: the best prescription for, say, four items may require four different drugs, although for any patient usually no more than two drugs need to be given (or should be, for reasons of drug interaction, toxic side effects, cost, etc.).

The original therapy program lacked a general scheme for relating the local constraints (best drug for each item) to the global constraint (fewest possible number of drugs). As we began to investigate the complexities of therapy selection, it became necessary to patch the program to deal with the special cases we encountered. Before long we were losing track of how any given change would affect the program's output. We found it increasingly difficult to keep records during the program execution for later use in the explanation system; indeed, the logic of the program was too confusing to explain easily. We decided to start over, aiming for a more structured algorithm that would provide sophisticated therapy, and by its very organization would provide simple explanations for a naive user. The question was this: what organization could balance these two, sometimes contradictory, goals?

Because we wanted to formulate judgments that could be provided *by* physicians and would appear familiar *to* them, we decided not to use mathematical methods such as evaluation polynomials or Bayesian analysis. On the other hand, MYCIN's inferential rule representation seemed to be inadequate because of the general algorithmic nature of the problem (i.e., iteration and complex data structures). We turned our attention to separating out the optimization criteria of therapy selection from control information (specifications for iteratively applying the heuristics). As is discussed below, the key improvement was to encode canonically the optimization performed by the inner loop of the algorithm.

6.2 Our Solution

6.2.1 Local and Global Criteria

We found that viewing the optimization problem in terms of local and global criteria provides a fruitful means for structuring the problem. Local criteria are the item-specific factors, such as sensitivity of the organism to preferred drugs, toxicity of drugs, the desire to "reserve" drugs for more serious diseases, and the desire to continue current therapy if possible. Global criteria deal with the entire recommendation; we wished to minimize the number of drugs, prescribing only two drugs if possible to cover for all of the most likely organisms.[1] In addition, there were a few patient factors to consider, such as allergies to antibiotics.

Besides providing for optimal therapy, we wished to provide for an explanation capability that would list simple descriptions of the therapy selection heuristics used by the algorithm, as well as reasons for not making a different recommendation.

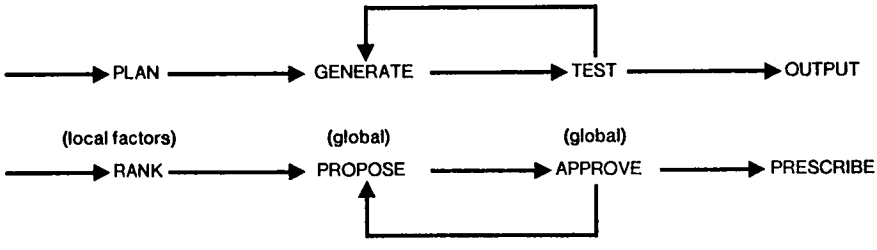

FIGURE 6-1 Therapy selection viewed as a plan-generate-and-test process.

After clearly stating these design goals, we needed an implementation scheme that would bring about the optimization. The key to our solution was the use of a generate-and-test control structure for separately applying the local and global factors. Figure 6-1 shows the steps of the plan-generate-and-test method and, below them, the corresponding steps of our algorithm. Briefly, the steps are

1. plan by ranking the drugs—the local factors are considered here;

[1]Here we realized that we could group the items into those that should definitely be treated ("most likely") and those that could be left out when three or more drugs would be necessary.

	Number of drugs of each rank:		
Instruction	*first*	*second*	*third*
1	1	0	0
2	2	0	0
3	1	1	0
4	1	0	1
.			
.			
.			

FIGURE 6-2 Instructions for the therapy proposer.

2. propose a recommendation and test it, thus dealing with the global factors; and
3. make a final recommendation.

The following sections consider these steps in more detail.

6.2.2 Plan

We start with an initial list of drugs to which each organism is sensitive and sort it by applying production rules for ranking. These reranking rules are applied independently for every organism to be treated. The chief purpose of this sorting process is to incorporate drug sensitivity information for the organisms growing in cultures taken from the patient.[2] Thus we arrive at a patient-specific list of drugs for each organism, reranked and grouped into first, second, and third ranks of choices.

Because this sorting process is a consideration specific to each organism, we refer to it as a local criterion of optimal therapy. We call it (loosely) a planning step because it makes preparations for later steps.

6.2.3 Generate

The second step of the algorithm is to take the ordered drug lists and generate possible recommendations. This is done by a proposer that selects subsets of drugs (a recommendation) from the collection of drugs for all of the organisms to be treated. Selection is directed by a fixed, ordered set of instructions that specify how many drugs to select from each preference group. The first few instructions are listed in Figure 6-2. For example, the

[2]A typical rule might be "If the organism growing from the culture appears to be resistant to the drug, then classify the drug as a third choice."

third instruction tells the proposer to select a drug from each of the first and second ranks. Instructions for one- and two-drug recommendations are taken from a static list; those for recommendations containing three or more drugs are generated from a simple pattern.

It should be clear that the ordering of the instructions ensures that two of the global criteria will be satisfied: prescribing one or two drugs if possible, and selecting the best possible drug(s) for each organism. An instruction therefore serves as a canonical description of a recommendation. Consequently, we can "reduce" alternate subsets of drugs to this form (the number of drugs of each rank) and compare them.

6.2.4 Test

Since all of the drugs for all of the organisms were grouped together for use by the proposer, it is quite possible that a proposed recommendation will not cover all of the most likely organisms. For example, the proposal might have two drugs that are in the first rank for one item but are second or third for other items, or are not even on their lists. Thus the first step of testing is to make sure that all of the most likely items are covered.

The second test ensures that each drug is in a unique drug class. For example, a proposal having both gentamicin and streptomycin would be rejected because these two drugs are aminoglycosides and therefore cause a "redundant" effect.

The last test is for patient-specific contraindications. These rules take into account allergies, age of the patient, pregnancy, etc. These rules are relatively expensive to apply, so they are done last, rather than applying them to each possible drug in the plan step. With this test we have dealt with the last global criterion of therapy selection. The first proposal that satisfies these three tests becomes the therapy advice. The details of drug prescription will not be considered further here; it consists primarily of algorithmic dosage calculation and adjustment in the case of renal failure.

6.2.5 Performance

We have found that the algorithm described above is manageable and performs well. It is straightforward to add new rules for ranking the drugs and for testing the proposals. The canonical instructions are relatively fixed, but it would not be difficult, for example, to provide infection-specific instruction sets. The program has made acceptable recommendations for a library of more than 100 meningitis patients.

6.3 The Explanation Capability

We will now consider how the structure of the algorithm is exploited to produce simple explanations. A sample question about therapy selection is shown in Figure 6-3. The medical decisions that were applied to the drug chloramphenicol are listed as a logical sequence of reasons, which is produced by retrieving and printing traces that were left behind by the program. The trace retrieval program is termed CHRONICLER because its explanations consist of a chronicle of decision events.

```
** WHY DID YOU GIVE CHLORAMPHENICOL FOR E.COLI IN REC-1?

CHLORAMPHENICOL was prescribed for ITEM-2 in RECOMMENDATION-1:

Since
     -- CHLORAMPHENICOL is a treatment of choice for e.coli in meningitis
     -- ITEM-2 is sensitive to CHLORAMPHENICOL
     -- there were no contraindications for it

CHLORAMPHENICOL was prescribed because it was part of the
recommendation that covers for all of the items, using the fewest number
of drugs.
```

FIGURE 6-3 A question concerning why a drug was prescribed. (User's input follows the double asterisks.)

Figure 6-4 shows the general organization of the Explanation System. The traces (discussed below) constitute a dynamic event history. A chronicle of events is printed by using a process transition diagram to selectively retrieve the relevant traces.

Figure 6-5 shows the kind of transition diagram we use to represent the steps of therapy selection. The states roughly correspond to the generate and test steps shown in Figure 6-1. The arrows are labeled as positive (pos) and negative (neg) criteria (i.e., criteria that support or oppose the recommendation of a given drug). These correspond to the medical strategies, e.g., "The drug is on the treatment-of-choice list for the organism (the initial list) and so was considered to cover for the organism." If a drug is prescribed, there must be a sequence of positive criteria leading from the first state to the output state. These are the reasons offered the user as an explanation for prescribing the drug. To make the explanation clearer, the states are reordered into three groups (planning criteria, testing criteria, and generate and output criteria) to conform to the following general scheme:

Since
--<plan criteria>
--<test criteria>

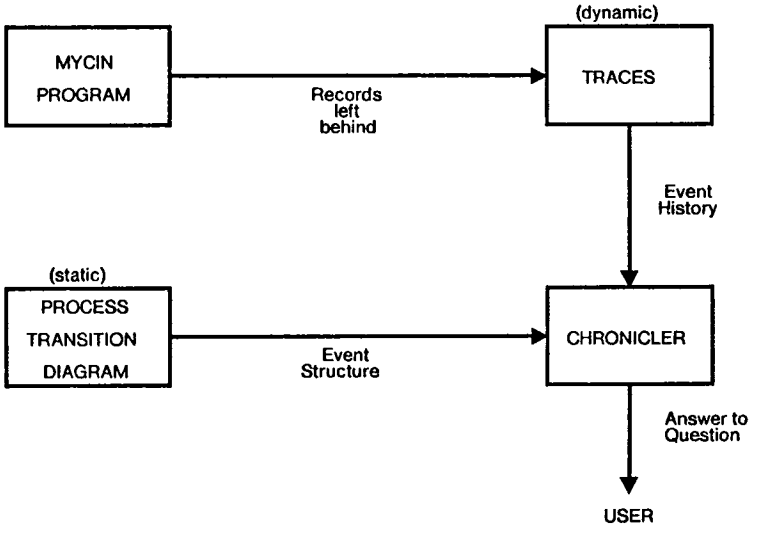

FIGURE 6-4 Organization of the Explanation System.

(therefore)
<generate and output criteria>

On the other hand, if a drug is not prescribed, there must be a negative criterion to explain why it dropped out of contention if it was on the initial list. Failure to prescribe can be caused by either failure to consider the

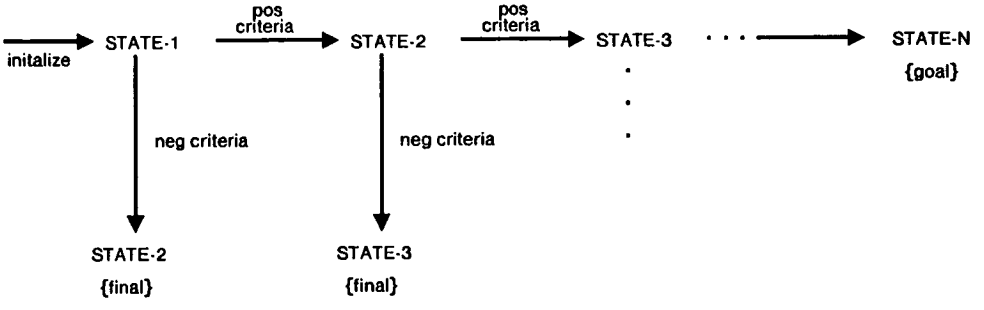

FIGURE 6-5 The state transition diagram.

**** WHY DIDN'T YOU SUGGEST PENICILLIN IN REC-1 FOR STAPH-COAG+?**

PENICILLIN was not prescribed for ITEM-1 in RECOMMENDATION-1:

PENICILLIN was discounted for ITEM-1 because it is NOT DEFINITE that the item is sensitive to this drug. There are other potential therapies under consideration which are much more desirable, viz., current therapies or drugs to which the item is definitely sensitive.

Would you like to see some details? **** YES**

The drugs to which the staphylococcus-coag-pos is sensitive are: cephalothin (1.0) vancomycin (1.0) gentamycin (1.0) tobramycin (1.0) erythromycin-and-tetracycline (1.0) chloramphenicol-and-erythromycin (1.0) [RULE098 RULE445]

Would you like to know about the history of PENICILLIN in the decision process up to this point? **** YES**

-- PENICILLIN is a treatment of choice for staphylococcus-coag-pos in meningitis. But as explained above, PENICILLIN was discounted.

FIGURE 6-6 Question concerning why a drug was *not* prescribed.

drug (plan) or failure of a test. A third possibility is that the drug wasn't part of an acceptable recommendation, but was otherwise a plausible choice (when considered alone). In this case, the drug needs to be considered in the context of a full recommendation for the patient.[3] (See Figure 6-9 for an example.)

Figure 6-6 shows an example of a question concerning why a drug was not prescribed. In response to a question of this type, the negative criterion is printed and the user is offered an opportunity to see the positive decisions accrued up to this point. In this example we see that penicillin was not prescribed because it is not definite that the item is sensitive to this drug. That is the negative criterion. The fact that penicillin was a potential treatment of choice permitted its transition to the reranking step.[4] This is shown in Figure 6-7. When MYCIN's rules (as opposed to Interlisp code) are used to make a transition decision, we can provide further details, as shown in Figure 6-6.

For questions involving two drugs, e.g., "Why did you prescribe chloramphenicol instead of penicillin for Item-1?", CHRONICLER is invoked to explain why the rejected drug was not given. Then the user is offered the opportunity to see why the other drug was given.

To summarize, MYCIN leaves behind traces that record the application

[3]Events are recorded as properties of the drugs they involve. The trace includes other contexts such as the item being considered. To deal with iteration, events are of two types: *enduring* and *pass-specific*. Enduring events represent decisions that, once made, are never reconsidered, e.g., the initial ranking of drugs for each organism. Pass-specific events may not figure in the final result; they may indicate computation that failed to produce a solution, e.g., proposing a drug as part of a specific recommendation. Thus traces are accessed by drug name and the context of the computation, including which pass of the generate-and-test process produced the final solution.

[4]Penicillin is given for staph-coag+ *only* if the organism is known to be sensitive to that agent.

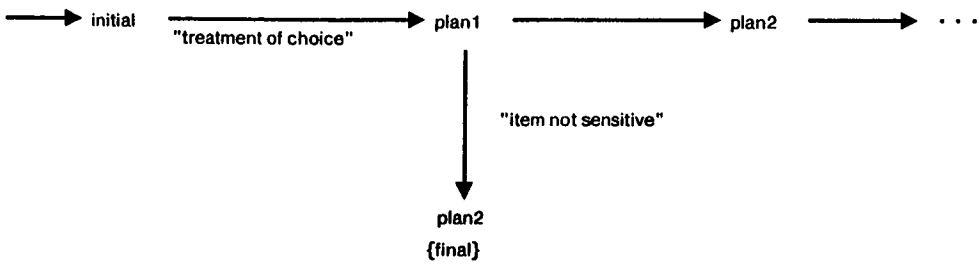

FIGURE 6-7 Trace history for the question shown in Figure 6-6.

of the positive and negative criteria. The Explanation System uses a state transition diagram that represents the steps of the algorithm to retrieve the relevant traces in a logical order.

It is interesting to note that CHRONICLER is described well by Bobrow and Brown's synthesis, contingent knowledge, and analysis (SCA) paradigm for understanding systems (Bobrow and Brown, 1975). Contingent knowledge is a record of program-synthesized observations for later use by an analysis program to answer questions or comment on the observed system. In CHRONICLER the traces and transition diagram constitute the contingent knowledge structure. Synthesis (abstraction of results) is performed by the therapy selector as it classifies the drugs in the various decision steps and records its "observations" in traces. Analysis is performed by CHRONICLER as it "reads" the traces, interpreting them in terms of the state transition diagram. The meteorology question-answering system described by Brown et al. (1973) uses a similar knowledge representation.

6.4 Comparing Alternative Recommendations

The therapy program will print alternative recommendations if asked to do so. This is done by eliminating drugs from contention and rerunning the algorithm. Figure 6-8 shows an example of an alternative recommendation and a comparison of this alternative to the algorithm's first choice. The comparison is made by simply reducing the recommendations to canonical form and then determining their relative ranking in the list of instructions. A recommendation is better than another when it reduces to a better instruction, i.e., when the preference description of its drugs appears earlier in the fixed list of instructions. It is also possible that the comparison

Do you wish to see the next choice therapy?
** **YES**
Which, if any, of the above drugs do you wish eliminated from consideration?
****CHLORAMPHENICOL**

[REC-2] Next best therapy recommendation:
In order to cover for Items <2 3 4>:
Give: GENTAMICIN

In order to cover for Item <1>:
Give: METHICILLIN

RECOMMENDATION-2 is substantially inferior to RECOMMENDATION-1 (because it requires more drugs or drugs that are not as good).

Would you like to see details of this comparison?
** **YES**

Looking at the coverage of the 'most likely' items, we see that:

RECOMMENDATION-1 consists of 1 first choice drug (CHLORAMPHENICOL for ITEM-2 and ITEM-3) and 1 second choice drug (GENTAMICIN for ITEM-3 and ITEM-1)

RECOMMENDATION-2 consists of 1 first choice drug (METHICILLIN for ITEM-1) and 1 third choice drug (GENTAMICIN for ITEM-2 and ITEM-3)

Do you wish to see the next choice therapy?
** **NO**

FIGURE 6-8 Comparing alternative recommendations.

will be a "close call," because one of the recommendations might use better drugs for the most likely organisms but cover for fewer of the less likely organisms. Again, it is the ability to encode output canonically that gives us the ability to make such a direct comparison of alternatives.

6.5 Evaluating a User's Choice of Therapy

The comparison described above is useful to a physician who prefers to give therapy other than MYCIN's first choice and wishes to know if the program truly considers it to be suboptimal therapy. However, it is tedious for the user to request all possible alternatives to be printed, so we offer the user the opportunity to enter his or her own choice of drugs for the organisms that require therapy (Figure 6-9).

Each drug the user suggests for an item is first formed into a standard internal question for CHRONICLER: "Why wasn't <drug> prescribed for <item>?" If there is a negative criterion about this drug for this item in the event history, it is printed and the user is given the option of selecting another drug.

**** WHY DIDN'T YOU GIVE AMPICILLIN-AND-GENTAMICIN FOR E.COLI IN REC-1?**

AMPICILLIN-AND-GENTAMICIN was not prescribed for ITEM-2 in RECOMMENDATION-1:

AMPICILLIN-AND-GENTAMICIN is a plausible choice for e.coli in meningitis, and was not explicitly rejected for use against ITEM-2 in RECOMMENDATION-1. However, the best therapy did not include AMPICILLIN-AND-GENTAMICIN.

If you would like to suggest therapy which includes AMPICILLIN-AND-GENTAMICIN, your regimen will be compared to MYCIN's. Would you like to do this? **** YES**

For each item in turn, enter the drug you would have prescribed in RECOMMENDATION-1.
In order to minimize the number of drugs in your recommendation, you may not want to prescribe therapy for every item. Items which represent the most likely organisms are indicated with a plus sign (+).

\+ ITEM-1 -- the staphylococcus-coag-pos **** GENTAMICIN**
\+ ITEM-2 -- the e.coli **** AMPICILLIN GENTAMICIN**
\+ ITEM-3 -- the klebsiella-pneumoniae **** GENTAMICIN**
\+ ITEM-4 -- the pseudomonas-aeruginose ******

[Checking for contraindications . . .]
[Considering AMPICILLIN-AND-GENTAMICIN for use against INFECTION-1 . . .]
[No contraindications found . . .]

[Now comparing your prescription to MYCIN's . . .]

Perhaps you did not realize that one of the drugs you prescribed, GENTAMICIN, will cover for ITEM-4, an item for which you did not prescribe therapy. I have changed your prescription accordingly.

ORGANISMS	Your regimen Drug -- Choice	MYCIN's regimen Drug -- Choice
"most likely"		
ITEM-3	GENTAMICIN -- 3rd	CHLORAMPHENICOL-AND-GENTAMICIN -- 1st
ITEM-2	AMPICILLIN-AND-GENTAMICIN -- 1st	CHLORAMPHENICOL -- 1st
ITEM-1	GENTAMICIN -- 2nd	GENTAMICIN -- 2nd
"less likely"		
ITEM-4	GENTAMICIN -- 2nd	GENTAMICIN -- 2nd

(The desirability of a drug is defined to be its lowest ranking for the items it covers.)

Both prescriptions include fewer than 3 drugs, so we must look at how highly ranked each prescription is for the most likely organism(s).

Your prescription of 1 first choice drug (AMPICILLIN for ITEM-2) and 1 third choice drug (GENTAMICIN for ITEM-3) is not as good as MYCIN's prescription of 1 first choice drug (CHLORAMPHENICOL for ITEM-2 and Item-3) and 1 second choice drug (GENTAMICIN for ITEM-1).

[You may refer to your regimen as RECOMMENDATION-2 in later questions.]

FIGURE 6-9 Evaluating a user's choice of therapy.

Once the user has supplied a set of drugs to cover for all of the most likely organisms, his or her proposal is tested for the criteria of drug class uniqueness and patient-specific factors (described in Section 6.2.4). If the proposal is approved, this recommendation is compared to the program's choice of therapy, just as the program compares its alternatives to its own first-choice recommendation.[5] It is also possible to directly invoke the therapy comparison routine.

6.6 Some Unsolved Problems

There are a number of improvements that could be made to this system. Among the most important to potential users is a more flexible question format. In our experience physicians tend to address short, unspecific questions to the program, e.g., "Why ampicillin?" or "What happened to E. coli?" Processing these questions will require a fairly sophisticated preprocessor that can help the user define such a question more precisely, or at least make some plausible assumptions.

Second, we anticipate the need to explain the heuristics, which now are describable only in a template form.[6] A user might like to know what a "drug sensitivity" is or why a heuristic was not used. Providing simple, fixed-text definitions is easy, but discussing a particular heuristic to the extent of explaining why it was not applicable is well beyond the capabilities of this Explanation System. One possible solution is to represent the heuristics internally in a rulelike form with a set of preconditions in program-readable predicates, like MYCIN's rules. We could then say, for example, that a drug was lowered in rank because its sensitivity was "intermediate," even though it was a current therapy (which would otherwise be reason for continuing to prescribe it). Thus we would be splitting a medical criterion into its logical components. Moreover, human explanations sometimes include hypothetical relations that have important instructional benefit, e.g., "If all of the drugs had been intermediate, then this current therapy would have been given preference." In general, paraphrasing explanations, explaining why an event failed to take place, and relating decisions are difficult because they require some representation of what the heuristics mean. Providing a handle on these underlying concepts is a far cry from a system that can only fill in templates.

Third, it is important to justify the medical heuristics and initial pref-

[5]The explanations at this point are more pedagogical than those supplied when the program compares its own alternatives. It seems desirable to phrase comparisons as positively as possible to avoid irritating the user.

[6]That is, each medical heuristic has a string with blanks associated with it, e.g., <drug> "was discounted for" <item> "because it was not *definite* that the item was sensitive to this drug."

erence ranks for drugs. We now provide text annotations that include references and comments about shortcomings and intent.

Finally, we could further develop the tutorial aspects of the Explanation System. Rather than passively answering questions, the Explanation System might endeavor to teach the user about the overall structure and philosophy of the program (upon request!). For example, a user might appreciate the optimality of the results better if he or she understood the separation of factors into local and global considerations. Besides explaining the results of a particular run, an Explanation System might characterize individual decisions in the context of the program's overall design. Parts Six and Eight discuss the issues of explanation and education in more detail.

6.7 Conclusions

We have developed a system that prescribes optimal therapy and is able to provide simple, useful explanations. The system is based on a number of design ideas that are summarized as follows:

1. separate the local and global optimality criteria;
2. apply these criteria in comprehensible steps—a generate-and-test control structure was found to be suitable;
3. justify selected therapies by using canonical descriptions that
   a. juggle several global criteria at once, and
   b. permit direct comparison of alternatives; and
4. exploit the simple control structure by using a state transition diagram to order retrieval of traces.

In addition, the Explanation System has benefited from a few simplifying factors:

1. There are relatively few traces (fewer than 50 drugs to keep track of and fewer than 25 strategies that might be applied).
2. There is a single basic question: Why was (or was not) a particular drug prescribed for a particular organism?

While this therapy selection algorithm may appear straightforward, it is the product of trying to codify an unstructured list of factors presented by physicians. The medical experts did not order these considerations and were not sure how conflicting constraints should be resolved. The framework we imposed, namely, invoking optimality criteria locally and globally

within a generate-and-test control structure and describing output canonically, provided a language that enabled us to codify the physicians' judgments, thereby significantly improving the performance and manageability of the program.

Moreover, this well-structured design enables us to print simple explanations of the program's decisions and to compare alternative solutions. We have provided this facility because we want the program to be used intelligently. If a user is confused or disagrees with the optimality criteria, we expect him or her to feel free to reject the results. The explanation system we have provided is intended to encourage thoughtful use of the therapy selection program.