

Chapter 2

Design Considerations for MYCIN[†]

2.1 Introduction

As discussed in Chapter 1, several computer programs have been written that attempt to model a physician's decision-making processes. Some of these have stressed the diagnostic process itself [Gorry, 1968a; Warner, 1972b; Wortman, 1972]; others have been designed principally for use as educational tools [Hoffer, 1973; Weinberg, 1973; Harless, 1973]; while still others have emphasized the program's role in providing medical consultations [Bleich, 1972; Peck, 1973; Kulikowski, 1973]. Actually, each of these applications is inherently interrelated since any program that is aimed at diagnosing disease has potential use for educating and counselling those who lack the expertise or statistical data that have been incorporated into the program. Consultation programs often include diagnosis as a major component although their principal focus involves interactive use by the physician and/or the determination of therapeutic advice.

In general, the educational programs designed for instruction of medical students and other health professionals have met with more long-term success [Wooster, 1973] than has been the case for the diagnostic and consultation programs. The relative success in implementing instructional programs may result because they deal only with hypothetical patients as part of an effort to teach diagnostic and therapeutic concepts, whereas the consultation programs attempt to assist the physician in the management of real patients in

[†]This chapter is based in large part on a paper presented by the author at the Thirteenth Annual San Diego Biomedical Symposium [Shortliffe, 1974a]. It is reproduced here with permission of the copyright owners.

MYCIN

the clinical setting. A program making decisions that can directly affect patient well-being must fulfill certain responsibilities to the physician if he is to accept the use of the computer and make use of its knowledge. This chapter discusses those clinical responsibilities and specifies the way in which they must be reflected in a system's design; specifically, the ways in which the MYCIN system seeks to satisfy these design considerations are described. Developmental concerns that relate to nonclinical criteria, such as economic, administrative, or legal requirements, are not included in this discussion.

2.2 Design Considerations for Consultation Programs

Physicians will, in general, reject a computer program designed for their use in decision making unless it is accessible, easy to use, forgiving of noncrucial errors from nonexpert typists, reliable, and fast enough to facilitate the physician's task without significantly prolonging the time required to accomplish it. They also require that the program function as a tool to the physician, not as an "all-knowing" machine that analyzes data and then states its inferences as dogma without justifying them.

Those who design computer programs to give advice to physicians should devise solutions to these requirements in an effort to combat the current lack of acceptance of computer-aided diagnosis by the medical profession [Croft, 1972]. The physician is most apt to need advice from such a program when an unusual diagnostic or therapeutic problem has arisen, precisely the circumstances under which the patient is likely to be acutely ill. Time will therefore often be an important consideration in such cases, and the physician may be unwilling to experiment with a program that does not meet the general requirements mentioned above.

With these considerations in mind, we developed the following list of prerequisites for the acceptance of a clinical consultation program. The list is idealistic, and its components are perhaps currently unattainable, but they do serve as useful guides as long-range goals for workers in the field. Each item is discussed in detail below, but a preliminary summary is presented here. In general, a therapeutic or diagnostic consultation program should be:

Design Considerations for MYCIN

- (1) useful;
- (2) educational when appropriate;
- (3) able to explain its advice;
- (4) able to understand and respond to simple questions stated in natural language;
- (5) able to acquire new knowledge, either through experience or by being told;
- (6) easily modified.

These design considerations are related to one another, and the need for each consideration tends to follow from those criteria listed above it. Furthermore, the order of development of capabilities occurs naturally from the bottom to the top of the list; for example, a program may not be able to explain its advice fully until it can respond to simple questions, and a program will not be useful until it can explain its advice. All six considerations, however, are aimed at satisfying those principles that reflect the system's responsibility to the physician and, through him, to the patient.

2.2.1 PROGRAM SHOULD BE USEFUL

Clearly the ultimate goal of any program is that it be "useful," and in the case of consultation systems for use by physicians this word has several important implications. Usefulness is measured along three scales:

- (1) the need for the assistance that the program provides;
- (2) the reliability of the advice;
- (3) the mechanics for accessing the machine and retrieving the desired information.

The validity of advice is of crucial importance. The system must give good advice most of the time and must be able to explain itself when it cannot reach a decision. Otherwise, physicians will soon learn that the system is of little practical value and will stop experimenting with it. Evaluative tests should demonstrate that the advice given by the program corresponds to that given by an expert who is provided with the same clinical information, or that the advice is, retrospectively, shown to be valid at least as often as is the advice of

MYCIN

the expert. This requirement means that the program must be given a large amount of knowledge before it is implemented on the hospital wards. In order to insure an accurate data base of clinical knowledge, cooperation and guidance from several experts in the field with which the program is involved is of great importance, and ongoing collaboration with physicians at all levels of system development is even more desirable. Practicing physicians tend to lose interest quickly in an experimental tool that is not clinically useful, even if they are warned that the program is still undergoing developmental work. It is therefore wise to defer implementation until the collaborating experts feel that minimal additional system improvement can be achieved prior to the ultimate test of ongoing clinical use.

The importance of "human-engineering" aspects of program design is often overlooked. Yet ignoring such issues can prevent acceptance of a system which otherwise gives good advice and fulfills the design criteria I have mentioned. In this sense a consultation program is not "useful" unless it is "useable." Doctors seek mechanisms for saving time without jeopardizing excellence of patient care, so a program that is slow, difficult to access, or frustrating to use will quickly be rejected. Once implemented, the system should be readily available to clinicians who may need its advice on short notice. Care should therefore be taken to provide a sufficient number of terminals so that there need not be lines of physicians waiting for their chance at the program. Furthermore, the user should require minimal training in order to get advice from the system. It is also desirable that system response time be fast and that the time from sign-on to sign-off be kept as short as possible commensurate with the difficulty of the therapeutic or diagnostic problem for which advice is being sought. If the program is not a multiple-choice or light-pen system and therefore requires typing by the physician, the amount of user input should be minimized and misspellings should be tolerated as much as possible. Users without computer experience tend to think that a machine is unintelligent if it cannot realize that "tetracyline" was intended to be "tetracycline," and a physician will not take kindly to a system that requires that he experiment with two or three spellings until he finds the one with which the program is familiar. Relatively minor issues such as these can make the difference between a successful consultation program, acceptable to clinicians, and one that is not. Minor issues should certainly not be

Design Considerations for MYCIN

ignored until clinical implementation is attempted because the problems can often be solved more easily if they are considered during program development.

2.2.2 PROGRAM SHOULD BE EDUCATIONAL WHEN APPROPRIATE

A physician who seeks advice from a therapeutic consultation program presumably recognizes that he may not have the necessary expertise or data to make the decision on his own. The program will therefore be interacting with an individual who is likely to welcome instructive comments regarding the patient and the way in which the specific therapeutic problem should be approached. However, the physician may not have time for a learning session with the machine. It is therefore not only important that the system be able to explain the knowledge required in order to make an appropriate clinical decision; it should also be sufficiently flexible so that it does not attempt to instruct the user unless requested to do so.

An additional benefit that accompanies the machine's ability to teach the user about its decision making is the possibility that, when similar clinical circumstances arise in the future, the physician will no longer need to turn to the consultation program. This can help avoid an over-dependence on the machine's capabilities.

2.2.3 PROGRAM SHOULD BE ABLE TO EXPLAIN ITS ADVICE

In most cases, the educational process I referred to above will be accomplished by having the machine explain the advice it has given. However, explanation serves more than an educational purpose. It also provides the program with a mechanism for justification of decisions; a physician will be more willing to accept a program's advice if he is able to understand the decision steps that the system has taken. This gives him a basis on which to reject the system's advice if he finds that the program is not able to justify its decisions sufficiently. It thereby helps the program conform to the physician's requirement that a consultation system be a tool and not a dogmatic replacement for the doctor's own decisions. Gorry has also discussed the need for explanation capabilities in diagnostic consultation systems [Gorry, 1973a] and suggests that the lack of such features in

MYCIN

Bayesian decision programs [Gorry, 1968a] partially accounts for their limited success when ward implementation has been attempted. Bleich attributes much of the success of his acid-base consultation program [Bleich, 1972] to its ability to discuss both the electrolyte status of the patient and its method for calculating the characteristics of the patient's acid-base disorder.

2.2.4 PROGRAM SHOULD BE ABLE TO UNDERSTAND QUESTIONS

A nonrestrictive mechanism by which the physician can communicate with the program is an important feature of a system designed to explain its decisions and educate the user. This is particularly true if an attempt is made to minimize specialized training for users of the program. Thus the program should be able to understand queries from the physician and it must be able to respond to requests for justification of decisions or machine-generated queries that may be puzzling. Yet few problems have given computer scientists more difficulty than the development of programs that can understand and act upon questions that are presented in natural language. As discussed in § 1.2.1-7, the field of computational linguistics has produced researchers who have approached natural language understanding from several different points of view [Schank, 1972; Winograd, 1972; Woods, 1970], and some investigators have dealt specifically with programs for understanding and answering questions [Simmons, 1970]. These programs have achieved results that are only of limited applicability. It is therefore unlikely that a consultation program developed for use in the clinical setting in the near future will have sophisticated natural language capabilities. Some attempt to solve the problem in a limited sense is appropriate at this time, however, since question-answering is a logical prerequisite for explanatory and educational capabilities.

2.2.5 PROGRAM SHOULD BE ABLE TO ACQUIRE NEW KNOWLEDGE

A program needs to be able to learn new information in any area of medical therapeutics where changes in decision criteria occur with

Design Considerations for MYCIN

some regularity. A facility for teaching new knowledge to the system is therefore desirable since expert clinicians are generally the only ones who can determine when the knowledge of the program is outdated or otherwise inadequate. The need for this kind of program reliability was discussed above. There is perhaps no better way to insure the reliability of the program's knowledge than to permit collaborating experts to experiment with the program during both developmental and implementation stages, to identify weaknesses in the system's decision criteria, and to make corrections or additions to the program's knowledge-base. After the program has been implemented in the clinical setting, a knowledge acquisition capability permits the system to continue to improve whenever errors in its decisions are found by an expert familiar with the methods for teaching it the necessary new information.

Realistically, however, few experts in medical therapeutics will have an extensive knowledge of computer programming and the inner workings of the consultation system. It is therefore important to enable the expert to teach the program new decision criteria or information by entering statements in English and letting the program interpret the language and determine how the new data should be incorporated into its knowledge-base. Although the computational problems involved are at least as difficult as those encountered during the question-answering task discussed above, this is a powerful capability that will greatly facilitate growth of the program's knowledge to a point at which the collaborating experts agree that the time for ward implementation has arrived.

A second kind of self-improvement by the program, and a feature that is more appropriate in some applications (such as therapy advisors) than in others, is the development of mechanisms for monitoring the effects of the system's advice upon patient welfare and for modifying its decision criteria dynamically in response to such observations. This kind of learning can take place only after implementation in the clinical environment has occurred and only if mechanisms exist for letting the machine know whether the physician has followed its advice and whether the patient has responded as desired to the medication that was administered. The issue should not be ignored during program development, however, because design of data structures and input/output mechanisms may be modified if the future need for such a facility is recognized.

MYCIN

2.2.6 PROGRAM KNOWLEDGE-BASE SHOULD BE EASILY MODIFIED

The need for straightforward system modification follows directly from the desire to permit the program to learn new information and decision criteria directly from the expert. If the teaching process requires intimate knowledge of the system's data base and how it is used, few clinical experts will have the time or inclination to acquire the necessary sophisticated insights into the program. For example, an inference model that depends on a complex decision tree is apt to be difficult to augment without a complete diagram of the tree so that all implications of additions can be observed. A modular system, on the other hand, permits knowledge to be acquired as isolated facts and allows the consultation program itself to decide under what conditions the new information is relevant. This requirement implies a great deal of intelligence in the consultation monitor but avoids the problems that result if the expert is asked to indicate exactly the circumstances under which the information he is offering may be useful.

Modularity of decision criteria also facilitates searches for inconsistencies or contradictions when new information is acquired during the learning process. If all system knowledge is stored in "packets," comparisons of a new "packet" with those that already exist can be straightforward. Such checks for contradictions are important if the system is to maintain its validity through many teaching sessions, particularly when several experts with different views of the consultation program's problem area are simultaneously influencing the system's knowledge.

It is possible that a consultation system can succeed to a certain extent without addressing itself to all of the design criteria just discussed or, on the other hand, that additional criteria need to be added to the list. However, the design considerations outlined in this chapter provide long-range goals that demand attention even for short-range program development since it is likely that the success of consultation programs will be impeded until each of these problems has been solved. MYCIN has been developed with all six design considerations in mind and, although it is not yet implemented for

Design Considerations for MYCIN

ongoing use in the clinical setting, it attempts to solve some of the serious design problems discussed above.

2.3 MYCIN and Acceptability Criteria

Several of MYCIN's interactive capabilities were demonstrated in the sample consideration included in § 1.5.2. In the remainder of this chapter I shall therefore present extracts of an interactive session, rather than an entire consultation, in an effort to point out how MYCIN reflects the six design considerations discussed above. Since the logical order of explanation of the six capabilities is from last to first, MYCIN's approach to each will be discussed in that order here. The programming details, however, will not be presented until Chapters 3, 5, and 6.

2.3.1 MODULARITY TO INSURE STRAIGHTFORWARD MODIFICATION

We accomplished modularity of system knowledge by storing all information in decision rules. These rules are coded in LISP internally, but can be translated into an English language version for communication with the user. For example, a rule that is presented to the physician as:

```
IF:      1) THE STAIN OF THE ORGANISM IS GRAMPOS, AND
         2) THE MORPHOLOGY OF THE ORGANISM IS COCCUS,
         AND
         3) THE GROWTH CONFORMATION OF THE ORGANISM
           IS CLUMPS
THEN:    THERE IS SUGGESTIVE EVIDENCE (.7) THAT THE
         IDENTITY OF THE ORGANISM IS STAPHYLOCOCCUS
```

is actually coded internally as:

```
PREMISE: ($AND (SAME CNTXT GRAM GRAMPOS)
              (SAME CNTXT MORPH COCCUS)
              (SAME CNTXT CONFORM CLUMPS))
ACTION: (CONCLUDE CNTXT IDENT STAPHYLOCOCCUS TALLY
        .7)
```

MYCIN

Since all system knowledge is stored in rules of this sort, and since there is no explicit reference to the way in which one rule relates to the others in the system, the decision rules maintain the modularity that is desirable. The way in which the program monitor (Subprogram 1 shown in Figure 1-1) decides which rules to use during an interactive session is explained in Chapter 3.

2.3.2 ABILITY TO ACQUIRE NEW KNOWLEDGE FROM EXPERTS

When the consultation program learns from the expert, the process is simply one of acquiring a new rule that conveys the information that the expert believes is lacking in the system. MYCIN's first 150 rules were determined over several months of meetings during which the collaborators discussed representative case histories. These rules were coded into LISP by hand and provided the core knowledge-base on which the early version of the consultation program has operated. Many future rules will be acquired interactively from infectious disease experts. Once an expert has determined what information is needed by the program, he indicates to the Rule-Acquisition System (Subprogram 3 shown in Figure 1-1) that he wishes to enter a new rule ("NR"). He is then asked to enter the rule in English, following the format of the other rules in the system. MYCIN attempts to translate the rule into a LISP representation and then uses its LISP-to-English routines to print out a version of the rule as it has understood the meaning. The expert is next asked to approve the new rule or to make modifications as necessary. The following session demonstrates this learning process (user input always follows the double asterisk):

****NR**

The new rule will be called RULE200.

If:	1	**THE ORGANISM IS A GRAM NEGATIVE ROD
and	2	**IT IS ANAEROBIC
and	3	**IT WAS ISOLATED FROM THE BLOOD
and	4	**YOU THINK THE PORTAL WAS THE GI TRACT
and	5	**

[user: carriage return with no entry]

Design Considerations for MYCIN

Then: 1 **IT IS PROBABLY A BACTEROIDES

On a scale of 1 to 10, how much certainty would you affix to this conclusion?

**9

and 2 **

[user: carriage return with no entry]

This is my understanding of your rule:

RULE200

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
3) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
4) THE AEROBICITY OF THE ORGANISM IS ANAEROBIC, AND
5) THE PORTAL OF ENTRY OF THE ORGANISM IS GI
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.9)
THAT THE IDENTITY OF THE ORGANISM IS BACTEROIDES

Okay? (YES or NO)

**YES

A similar procedure is used to alter previously existing rules that the expert finds incomplete or otherwise inadequate. Only after expert clinicians have used these features to teach MYCIN additional knowledge will the validity of the program's advice reach an adequate level so that implementation in the hospital can be considered.

The mechanism that MYCIN uses for understanding the natural language input of experts avoids syntactic or semantic analysis; it is, rather, a modified key-word approach in which accumulated clues are combined in order to deduce which attributes of organisms, cultures, patients, etc., are being discussed. The technique is further described in Chapter 5.

Once new rules are acquired from the expert, they immediately become available for use by the Consultation System (Subprogram 1). Appropriate checks must be made to insure that the rule does not contradict any previously existing rule, but the expert need not worry about how the rule interrelates with other rules or when the system should use the new knowledge. At present, only a limited

MYCIN

number of rules have been acquired using the scenario demonstrated above (see § 6.3). Future work will concentrate on the problem of letting MYCIN learn rules that discuss attributes with which the system is not already familiar. In such instances, the program's 800-word dictionary is of small value in deducing the meaning of the English-language input and new techniques must be developed so that MYCIN can not only learn new rules of this variety but can also learn about the clinical attributes that are being discussed.

2.3.3 ABILITY TO UNDERSTAND QUESTIONS

MYCIN answers questions about its decisions by retrieving and printing out the relevant rules (Subprogram 2 shown in Figure 1-1). Much of the intelligence of the program thus lies in the portion of the program which finds these "relevant" rules. Questions are answered within 5 to 20 seconds, considerably faster than is the case with more complex natural language question-answering systems in which both syntactic and semantic content must be analyzed. A physician may ask either (a) informational questions (prefaced with the letters "IQ") which query the status of current knowledge about the patient, (b) questions about the deductions of the current consultation, or (c) general questions about any of MYCIN's judgmental rules, whether used in the current consultation or not. The following question-answering examples demonstrate these capabilities:

(a) Informational question:

**IQ WHAT IS THE IDENTITY OF ORGANISM-1?

THE IDENTITY OF ORGANISM-1 IS E.COLI (.43) PROTEUS (.22)
PSEUDOMONAS (.22) KLEBSIELLA (.13)

(b) Questions about the current consultation:

**HOW DID YOU DECIDE THAT ORGANISM-1 MIGHT BE A

**PSEUDOMONAS?

I USED RULE085 TO CONCLUDE THAT THE IDENTITY OF
ORGANISM-1 IS PSEUDOMONAS. THIS GAVE A CUMULATIVE
CERTAINTY FACTOR OF .6

Design Considerations for MYCIN

(c) General questions:

****PR RULE085**

RULE085

IF: 1) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
 2) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
 3) THE PATIENT IS A COMPROMISED HOST
THEN: THERE IS SUGGESTIVE EVIDENCE (.6) THAT THE
 IDENTITY OF THE ORGANISM IS PSEUDOMONAS

****DO YOU EVER PRESCRIBE CARBENICILLIN FOR PSEUDOMONAS
INFECTIONS?

Relevant rules: RULE024 RULE068 RULE137 RULE138

Which ones do you wish to see?

**** RULE024**

RULE024

IF: 1) THE THERAPY UNDER CONSIDERATION IS
 GENTAMICIN, AND
 2) THE IDENTITY OF THE ORGANISM IS DEFINITELY
 PSEUDOMONAS
THEN: RECORD THE FOLLOWING AS A MORE APPROPRIATE
 THERAPY THAN GENTAMICIN ALONE: THE
 COMBINATION OF GENTAMICIN AND CARBENICILLIN

It is also possible to ask the system (d) to explain questions in retrospect (e.g., "Why did you ask question 17?") and (e) to ask for confirmation of one's own decision rules (e.g., "Is chloramphenicol okay for Salmonella infections?"). The implementation details for these capabilities are the subject of Chapter 5.

2.3.4 ABILITY TO EXPLAIN DECISIONS

It should be clear from the above explanation session that the ability to answer questions and remember the details of a consultation provide the mechanism for explaining decisions which may puzzle the user. Questions of type (b) require that the machine refer to the specific details of the consultation it has just undertaken in order to justify its final decisions. This explanation technique allows

MYCIN

the physician to choose those points which he feels need clarification and permits him to avoid questions altogether if he finds the system's advice consistent with his own opinions regarding therapy for the patient.

The question-answering session usually follows a consultation session. However, questions may occur to the physician during the consultation itself and we have therefore implemented two additional explanatory capabilities. One of these allows the user to enter "QA" in response to any question that is asked by MYCIN. This tells the system to digress temporarily from the consultation and to permit the user to ask questions of the type demonstrated in § 2.3.3. In this way, the user can query partial results of the program's reasoning and then return control to MYCIN (by entering the word STOP) so that the consultation may proceed from the point of digression.

The second capability permits the user to demand that MYCIN justify any question that is asked. Whenever a question generated by MYCIN puzzles the physician, he simply enters the word RULE and the program responds by printing out the translation of the decision rule that has generated the current question. After printing out the relevant decision rule, MYCIN repeats its question and the consultation continues unhampered. For example:

17) Are there any factors in this patient which may cause inadequate gastro-intestinal absorption of the penicillin (DRUG-1)?

**RULE

RULE049

IF: 1) THE ROUTE OF ADMINISTRATION OF THE
 PENICILLIN IS ORAL, AND
 2) THERE IS A GI FACTOR WHICH MAY INTERFERE
 WITH THE ABSORPTION OF THE PENICILLIN
THEN: THERE IS SUGGESTIVE EVIDENCE (.6) THAT THE
 ROUTE OF ADMINISTRATION OF THE PENICILLIN IS
 NOT ADEQUATE

17) Are there any factors in this patient which may cause inadequate gastro-intestinal absorption of the penicillin (DRUG-1)?

**NO

A similar capability, the WHY option, provides a more detailed and conversational explanation of the program's reasoning. A discus-

Design Considerations for MYCIN

sion of the complexities of implementation of this and related capabilities, plus a description of their use, may be found elsewhere [Shortliffe, 1975b; Davis, 1976].

2.3.5 EDUCATIONAL CAPABILITIES

As was pointed out in the discussion of the six design considerations, the ability of a consultation program to explain its decisions, and to answer questions about the area of expertise that it is modeling, automatically provides an educational capability. The sample question-answering session and the RULE option demonstrate the variety of ways in which MYCIN educates the user as well as justifies its decisions.

2.3.6 GENERAL USEFULNESS

As has already been stated, the ultimate test of MYCIN's usefulness and acceptability will come when we finally feel it is ready to be installed in the ward setting. In an effort to prepare for that day, we have tried to develop interactive characteristics that will overcome the standard complaints voiced by physicians who try to use terminal-based systems.

Whenever MYCIN asks a question, it knows the range of possible answers. It therefore compares the physician's response against the list of recognized responses. If the user's response is not on the list, it determines whether a simple typographical or spelling error will account for the unrecognized response. If spelling correction is unsuccessful, the system lexicon is checked to see if the user's answer is a synonym for one of the recognized responses. If this attempt fails, MYCIN prints out a list of recognized responses and asks the question again.

Both spelling-correction and the listing of recognized responses help reduce the level of frustration that can easily alienate novice users of computer systems. Additional features have also been implemented to assist the physician when he is puzzled by a question that MYCIN is asking. If he enters a question-mark ("?"), MYCIN assumes that he would like to see some sample responses. In addition, any question can be answered with the letters UNK (for UNKnown) if the user is uncertain of the answer but wishes MYCIN's opinion in

MYCIN

spite of the incomplete information. Finally, the **RULE** and **WHY** options that have already been mentioned help the user feel comfortable with the system and more inclined to accept MYCIN as the clinical tool it is designed to be.

This chapter has concentrated on explaining *why* the MYCIN system operates the way that it does. The next three chapters will deal with a description of *how* these goals have been accomplished. In Chapter 3 the subject is the core consultation program itself (Subprogram 1). Chapter 4 explains the mechanism we have devised for quantification of the program's decision processes. Then Chapter 5 summarizes MYCIN's question-answering capabilities (Subprogram 2). The program's limited ability to learn from experts (Subprogram 3) is included in § 6.3 when I discuss future efforts contemplated for improving MYCIN's acceptability and for extending its range of uses.