PART ELEVEN

Designing for Human Use

32

Human Engineering of Medical Expert Systems

Although we have frequently referred to human engineering issues throughout this book and have considered them from the outset in our design of MYCIN and its descendents, we have also noted that MYCIN was never used routinely in patient-care settings. Yes, the program was able to explain its reasoning, and this seemed likely to heighten its acceptability. And yes, we spent much time attending to detail so that (a) user aids were available at any time through the use of HELP and question mark commands, (b) the system automatically corrected spelling errors when it was "obvious" what the user meant, and (c) a physician could enter only the first few characters of a response if what was entered uniquely defined the intended answer. However, there were still significant barriers that prevented us from undertaking the move to formal implementation.

Some of these barriers were unrelated to human engineering issues, viz., the need for an enhanced knowledge base in other areas of infectious disease at a time when both Axline and Yu were departing from Stanford, the difficulty of obtaining funding for knowledge base enhancement when the program itself had become both large and competent, and our own lack of enthusiasm for implementation studies once we had come to identify some of the computer science inadequacies in MYCIN's design and preferred to work on those in a new environment. All of these might have been ignored, however, since MYCIN was fully operational and could have been tested clinically with relatively little incremental effort. What dissuaded us from doing so was the simple fact that we *knew* the program was likely to be unacceptable, for mundane reasons quite separate from its excellent decision-making performance. Most of these issues were related to logistical and human-engineering problems in the program's introduction. We have described these pragmatic considerations elsewhere (Shortliffe, 1982a) and have indicated how they influenced our decision to turn our attention to the development of a new system for clinical oncology (see Chapter 35). We will briefly summarize these points here.

First, although there was a demonstrated need for a system like MYCIN (see the data on antibiotic use outlined in Chapter 1), we did not feel there was a *recognized* need on the part of individual practitioners. Most physicians seem to be quite satisfied with their criteria for antibiotic selection, and we were unconvinced that they would be highly motivated to seek advice from MYCIN, particularly in light of the other problems noted below.

Our second concern was our inability to integrate MYCIN naturally into the daily activities of practitioners. The program required a special incremental effort on their part: once they had decided to consider giving a patient an antibiotic, it would have been necessary to find an available terminal, log on, and then respond to a series of questions (many of which were simply transcriptions of lab results already known to be available on other computers at Stanford). Linkage of SUMEX (MYCIN's "home" computer) to Stanford lab machines was considered but rejected because of lack of resources to do so and the realization that a research machine like SUMEX would still have been unable to offer high-quality reliable service to physician users. When the machine was heavily loaded, annoying pauses between MYCIN's questions were inevitable, and a total consultation could have required as long as 30 minutes or an hour. This was clearly unacceptable and would have led to rejection of the system despite its other strong features. Slight annoyances, such as the requirement that the physicians type their answers, would have further alienated users. Adapting MYCIN to run on its own machine was an unrealistic answer because of the computational resources needed to run a program of that size (at that time) and our lack of interest in trying to adapt the code for a non-Interlisp environment.[1]

Thus, as of late 1978, MYCIN became a static system, maintained on SUMEX for demonstration purposes and for student projects but no longer the subject of active research. In addition, in the subsequent five years its knowledge base has become rapidly outdated, particularly with regard to antimicrobial agents. The "third-generation" cephalosporins have been introduced in the intervening years and have had a profound effect on antibiotic selection for a number of common problems in infectious disease (because of their broad spectrum and low toxicity relative to older agents). This point emphasizes the need for knowledge base maintenance mechanisms once expert systems are introduced for routine use in dynamic environments, where knowledge may change rapidly over time.

Even though MYCIN is no longer a subject of active work, the experiments described in this book have been a productive source of new insights. In this final section to the book, we describe related pieces of work that show some of the ways in which MYCIN has influenced our research

---

[1]The CONGEN program within DENDRAL had just been recoded from Interlisp to BCPL, and we were acutely aware of the manpower investment it took by someone intimately familiar with the design and code. This effort could only have been undertaken under the conviction that the result would be widely used.

activities in the areas of human engineering and user attitudes. Our new work on ONCOCIN, for example, has been based on underlying knowledge structures developed for MYCIN but has been augmented and revised extensively because of our desire to overcome the barriers that prevented the clinical implementation of MYCIN. Our attitude on the importance of human factors in designing and building expert systems is reflected in the title of a recent editorial we prepared on the subject: "Good Advice is Not Enough" (Shortliffe, 1982b).

32.1 The Interface Language for Physicians

It was never our intention to become enmeshed in the difficult problems of understanding unconstrained English. Work in computational linguistics achieved important results during the 1960s and 1970s, but we saw the problems as being extremely difficult and were afraid that our progress in other areas would be slowed if we became overly involved in building language capabilities for MYCIN. We *did* spend time ensuring that the program could express itself in English, but this was not difficult because of the stereotypic form of the rules and the power of LISP. We totally avoided any need for the program to understand natural language during the consultation (depending instead on HOW, WHY, and EXPLAIN commands as described in Chapter 18), but we did build a simple question-answering (QA) system that was available electively at the end of the advice session. Although it was possible to get answers to most questions using the QA module, the system was not very robust, and it took new users some time to learn how to express themselves so that they would be understood. Once again, the capability that was developed for question answering (which was borrowed for the TEIRESIAS work; see Chapter 9) was greatly facilitated by the highly structured and uniform techniques for knowledge representation that we had used.

It is important to note that our desire to avoid natural language processing accounts in large part for the decision to use goal-directed (backward-chained) reasoning in MYCIN. If we had simply allowed the user to start a consultation by describing a patient, it would have been necessary that MYCIN understand such text descriptions before beginning forward-chained invocation of rules. By using a backward-chained approach, MYCIN controlled the dialogue and therefore could ask specific questions that generally required one- or two-word answers.

From a human-engineering viewpoint, this decision was suboptimal, even though, ironically, it was made to avoid language-understanding problems that we knew would have annoyed physician users. The problem that resulted from having MYCIN control the dialogue was the inability of the user to volunteer information, meaning that he or she had to wait for MYCIN to ask about what was known to be a crucial point. Alain Bonnet, a postdoctoral fellow from France, was fascinated by this problem when he visited our group in the mid-1970s. He decided to look for ways in which MYCIN's knowledge structures could be augmented to permit volunteered information about a patient at the beginning of a consultation session. His work on this subsystem, known as BAOBAB, is described in Chapter 33. The complexity of the issues that needed to be addressed in building such a capability are clear in that article. Fascinating though the work was, BAOBAB never functioned at a performance level sufficiently high to justify its incorporation into MYCIN.

Despite the limitations of its language capabilities, we are generally pleased with the ability of MYCIN and the EMYCIN systems to appear to converse in English through the use of rather simple techniques of text generation and understanding. This conversational appearance of the program is due to the combined efforts of several project members and to the flexibility of the underlying knowledge structures used. Issues in computational linguistics in the EMYCIN environment continue to be fruitful areas of investigation for student projects. As recently as 1980, a medical student and research assistant, Lou Sanner, added code to MYCIN that was able to generate prose summaries of patients from our library of old cases. His generalized approach to the problem was added to EMYCIN and generates prose descriptions of stored cases from any EMYCIN domain. An example of one of his MYCIN case translations is shown in Figure 32-1.

32.2 Assessing Physicians' Attitudes

As many of the early papers in this volume indicate, we proceeded through the 1970s with the firm conviction that AI techniques offered potential solutions to problems that had limited physicians' acceptance of advice-giving systems. We were especially convinced that explanation capabilities were crucial for user acceptance and that this single failing in particular largely accounted for the rejection of systems based solely on statistical approaches. As is discussed in Chapter 30, we could not *prove* that explanations would make a difference unless we implemented a consultation system in a clinical environment where controlled studies could be undertaken. Thus we had depended on our intuitions and appealed to others to believe in what we felt was an obvious requirement for optimal systems.

In 1980, however, a combination of events encouraged us to undertake a formal analysis of physicians' attitudes. We had toyed with the idea for several years but had been discouraged by the time and resources necessary

A summary is now being generated:
[consultation of 7-May-77 6:00PM]

Pt600 is a 33 year old Caucasian female with clinical evidence of otitis media who has neurological signs of 5 hours and symptoms of 1 day duration. She is febrile and weighs 70 kgm. She has impaired renal function. She is 4 + sick (on a scale of 4). The patient is thought to have a csf infection symptomatic for 1 day.

TEST RESULTS:

CBC:	WBC	25K		PMNS	85%	Bands	12%
CSF:	WBC	12500		PMNS	98%		
	glucose		25		(blood glucose 140)		
	protein	450					
recent serum creatinine 1							

CULTURES:	When obtained:	Organisms
csf	6 hours ago	Gramneg rod Grampos coccus in pairs

DRUGS:

Erythromycin was started (oral) 30 hours ago.

FIGURE 32-1 Example of a MYCIN case summary.

to do such a study well. In August of 1980 Stanford hosted the annual Workshop on Artificial Intelligence in Medicine, and we organized a two-day tutorial program so that local physicians who were interested could learn about this emerging discipline. In addition, funding from the Henry J. Kaiser Family Foundation allowed us to support a questionnaire-based project to assess physicians' attitudes. Finally, a doctoral student in educational psychology, Randy Teach, joined the project that summer and brought with him much-needed skills in the areas of statistics, study design, and the use of computer-based statistical packages.

The resulting study used the physicians who were attending the AIM tutorial as subjects, with a control group of M.D.'s drawn from the surrounding community. Chapter 34 summarizes the results and concludes with design recommendations derived from the data analysis. The reader is referred to that chapter for details; however, it is pertinent to reiterate here that a program's ability to give explanations for its reasoning was judged to be the single most important requirement for an advice-giving system in medicine. This observation accounts for our continued commitment to research on explanation, both in the ONCOCIN program (Langlotz and Shortliffe, 1983) and in current doctoral dissertations from the Heuristic Programming Project (Cooper, 1984; Kunz, 1984). Other results of the attitude survey reemphasize the importance of human-engineering issues (such as ease of use and access) in the design of acceptable consulting systems.

32.3 Clinical Implementation of an Expert System

It seems appropriate that we close a book about the MYCIN "experiments" with a description of ONCOCIN, MYCIN's most recent descendent. The problem domain for this program was selected precisely because it seemed to offer an excellent match between the problem-solving task involved and the set of pragmatic considerations that we outlined at the beginning of this chapter. Chapter 35 describes ONCOCIN's task domain in some detail and discusses the knowledge structures and architecture used to heighten its clinical effectiveness. However, Chapter 35 does not discuss the logistics of implementation that are among the newest lessons learned by our group. Thus what follows here is a description of our experience with ONCOCIN's implementation. Much of the discussion is drawn from a recent paper written by members of the ONCOCIN project (Bischoff et al., 1983). The reader may find it useful to study the technical description in Chapter 35 before reading this discussion of what has happened since the system was introduced for clinical use.

ONCOCIN assists physicians with the management of patients enrolled in experimental plans (called protocols) for treating cancer with chemotherapy. The system has been in limited use in the Stanford Oncology Clinic since May of 1981. The potential utility of such a system has been recognized at several major cancer treatment centers, and other groups have been developing systems to assist with similar tasks (Horwitz et al., 1980; Blum et al., 1980; Wirtschafter et al., 1980). Since the core of knowledge about oncology protocols is defined in protocol documents, the domain of cancer chemotherapy has the advantage of having a readily available source of structured knowledge of the field. The ongoing involvement of oncologists with ONCOCIN, both as research colleagues and as potential users, has provided additional expertise and highly motivated collaboration in knowledge base development. We currently have encoded the protocols for Hodgkin's disease, non-Hodgkin's lymphoma, breast cancer, and oat cell carcinoma of the lung[2] and will be adding all of the other treatment protocols employed at Stanford. It should be emphasized that the resulting computer-based protocols include both the specific rules gleaned from the protocol documents *and* some additional judgmental expertise from our experts, who have defined the ways in which the system ought to respond to unusual or aberrant situations.[3]

[2]The oat cell protocol is the most complex protocol at Stanford. It was implemented to verify that our representation scheme would apply to essentially any of the protocols currently in use. However, it has not yet been released for routine use, pending its thorough testing.

[3]In order to design a program that could be operational in the short term, our initial design plan was consciously to avoid major theoretical barriers such as management of inexact reasoning and generalized methods for temporal reasoning.

32.3.1 System Design

ONCOCIN's system design is a result of the combined efforts of an interdisciplinary group of computer scientists, clinicians, statisticians and support staff, totaling 29 individuals. System design began in July of 1979. From the outset, the logistics of how a consultation system could fit into the busy oncology clinic were a crucial design consideration; one of our first tasks was to study the flow of information within the clinic. We asked the oncology fellows about their attitudes regarding computers and asked them to assess the potential role of such technology in the oncology clinic. A Stanford industrial engineer with experience in the area of human factors was consulted during the iterative phase of interface design. Programmers would offer mock demonstrations to those with little or no computer expertise. After getting comments and suggestions on the demonstration, modifications were made, and a new mock-up was presented. This process was repeated until all felt satisfied with the interaction. Design decisions of this type were discussed at regular research meetings involving both physicians and computer scientists.

The design of the reasoning program, which is written in Interlisp and uses AI representation techniques (see Chapter 35), was affected by our desire to create a system that provides rapid response. The original ONCOCIN prototype used keyboard-oriented interactive programs borrowed from EMYCIN. As was mentioned earlier in this chapter, we knew from our previous work, however, that this type of interaction would be too tedious and time-consuming for a busy clinic physician. A physician using MYCIN often had to wait while questions were generated and rules were tried. The use of the EMYCIN interface, however, enabled us to create the program's knowledge base and to evaluate its therapy recommendations while we were concurrently deciding on the interface design. The ultimate interface incorporates a fast display program that is separate from the AI reasoning program (Gerring et al., 1982). Thus ONCOCIN is actually a set of independent programs that run in parallel and communicate with each other.

A major design goal was to have ONCOCIN used directly by physicians at the time of a patient's visit to the clinic for chemotherapy. One way to encourage physicians' involvement was to make the system easily accessible while providing a variety of hard-copy reports that had previously either not existed or required manual preparation. A computer-generated summary sheet is produced in the morning for each scheduled patient enrolled in one of the protocols handled by the computer. The summary sheet is attached to the patient's chart and serves as a reminder of the patient's diagnosis and stage, expected chemotherapy, and any recent abnormal laboratory values or toxicities. A centrally located video display terminal is used by the oncologist after the patient has been examined. The physician interacts with ONCOCIN's high-speed data acquisition program (the *Interviewer*). While the clinician is entering data through the Interviewer, that program is passing pertinent answers to the reasoning program (the *Reasoner*), which uses the current patient data, the past history, and the protocol assignment to formulate a treatment plan. By the time data entry is complete, the Reasoner has generally completed its plan formulation and has passed the results back to the Interviewer, which in turn displays the recommendation to the user. The physician can then agree with or modify the system's treatment recommendation, make adjustments to the laboratory and x-ray tests suggested for the patient by ONCOCIN, and end the session. Progress notes are produced on a printer near the ONCOCIN terminal so they can be easily removed, verified and signed by the physician, and then placed in the hospital chart. After the session the computer also generates an *encounter sheet,* which lists the tests to be ordered, when they should be scheduled, and when the patient should return to the clinic for his or her next visit. This information is generated on a second printer located at the front desk, where these activities are scheduled.

The system design attempts to prevent the computer system from being perceived as an unwanted intrusion into the clinic. The physician/computer interaction takes the place of a task that the physician would otherwise perform by hand (the manual completion of a patient flow sheet) and requires only 5 to 7 minutes at the terminal. A training session of 30 minutes has been adequate for physicians to achieve independent use of the system, and the hard-copy reports assist the physicians with their responsibilities. Because we were eager to make the system as flexible as possible and to simulate the freedom of choice available to the physicians when they fill out the flow sheets by hand, the program leaves the users largely in control of the interaction. Except for the patient's white cell count, platelet count, and information about recent radiation therapy (key issues in determining appropriate therapy), the physicians may enter whatever information they feel is pertinent, leaving some fields blank if they wish. An important evaluative issue that we are accordingly investigating is whether ONCOCIN encourages more complete and accurate recording of the flow sheet data despite the user's ability to skip entries if he or she wishes to do so. Users may enter data into the flow sheet format in whatever order they prefer, skipping forward or backward and changing current or old answers. This approach is radically different from that used in MYCIN in that the physician decides what information to enter and the reasoning can proceed in a data-directed fashion. Data entry in a flow sheet format avoids the problems of natural language understanding that prevented this approach in MYCIN.

32.3.2 Terminal Interface

The system incorporates a special terminal interface to ensure that a busy clinician can find ONCOCIN fast and easy to use, as well as simple to learn. The physician interacts with a high-speed (9600 baud) video display ter-

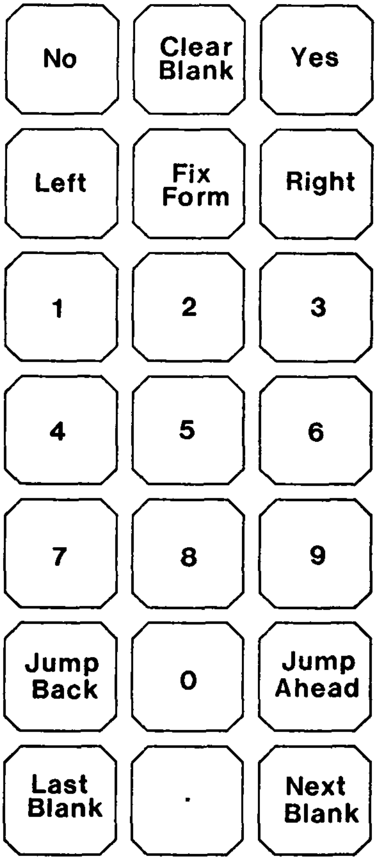

FIGURE 32-2 ONCOCIN's 21-key pad.

minal with multiple windows, simulating the appearance of the conventional paper flow sheet. Simulation of the form makes the interaction more comfortable and familiar.

A customized keyboard was designed for data entry. It allows the physician to enter the flow sheet information using a 21-key pad (Figure 32-2), which is located to the right of a conventional terminal keyboard. We considered light pens and touch screens but felt that they were either too expensive or too unreliable at the present time. Furthermore, a simple key pad was adequate for our needs. The layout of the key pad is simple and self-explanatory. Ten of the keys make up a number pad, which is laid out the same way as the numbers on push-button telephones. Our human factors consultant recommended this arrangement because we could safely assume user experience with push-button telephones, while user experience with a calculator-style number pad would be likely to be more limited. The other keys on the pad are "Yes" and "No" keys, and cursor control keys. The labels on the cursor control keys suggest that the user is filling in the blanks on a paper form, for example, "Next Blank," "Clear Blank," "Jump Ahead," etc. Our human factors consultant suggested using this terminology instead of terms including the word "Field" (e.g., "Next Field"), which are information-processing terminology and not as intuitive for naive computer users. This decision reflects our general effort to avoid computer jargon in talking with physicians, printing text on the terminal screen, or communicating with them in memos.

32.3.3 Display Design

The design of the display is derived from the paper flow sheet used for many years for protocol data gathering and analysis. The display screen is divided into four sections as indicated in Figure 32-3:

a. the *explanation field,* which presents the justification for the recommendation indicated by the user-controlled cursor location (the black block in the figure)
b. the *message field,* which identifies the patient and provides a region for sending pertinent messages from ONCOCIN to the physician
c. the *flow sheet,* which displays a region of the conventional hard copy flow sheet; the display includes columns for past visits, and the physician enters data and receives recommendations in the right-hand column
d. the *soft key identifiers,* labels that indicate the special functions associated with numbered keys across the top of the terminal keyboard

Note that when the physician is entering patient data, the explanation field specifies the range of expected entries for the item with which the cursor is aligned. When the system has recommended therapy (as in Figure 32-3), the explanation field provides a brief justification of the drug dosage indicated by the cursor location.

32.3.4 Integration into the Clinic

To make ONCOCIN's integration into the clinic as smooth as possible, we scheduled clinic meetings led by the oncology members of our research team. At one early meeting to announce that the system would soon be available, we gave a system demonstration and held a discussion of our project goals. Individual training sessions were then scheduled to teach each physician how to use the system. These orientation sessions were brief and informative. They stressed that the physician is the ultimate decision

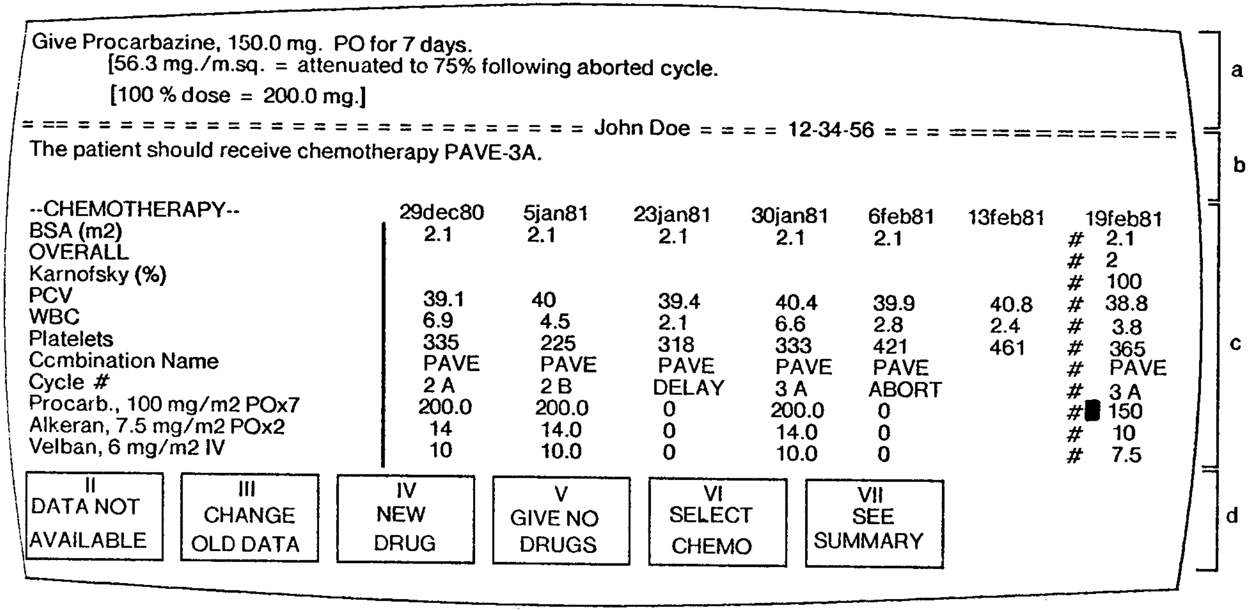

FIGURE 32-3 Sample ONCOCIN screen.

maker about the patient's care, and that the computer-based consultant is intended to remind the physician about the complex details of the protocols and to collect patient data. Members of our group meet with oncology faculty and physicians occasionally to give them progress reports on our research.

We also enlisted the help of a data manager who is responsible for training sessions, ensures that on-line patient records are current, and sees that the system runs smoothly. The data manager is available whenever the system is running in the clinic and offers assistance when necessary. This role has proved to be particularly crucial. The data manager is the most visible representative of our group in the clinic (other than the collaborating oncologists themselves). The person selected for this role therefore must be responsible, personable, tactful, intelligent, aware of the system's goals and capabilities, and able to communicate effectively with the physicians. If the person in this role is unable to satisfy these qualifications, he or she can make system use seem difficult, undesirable, and imposing to the physician users.

Integration of the system into the clinic was planned as a gradual process. When the system was first released, the program handled a small number of patients and protocols. As the program became more familiar to the physicians, we added more patients to the system. We are in the process of adding new protocols, which in turn will mean additional patients being handled on the computer. ONCOCIN was initially available only three mornings per week. It is now available whenever patients who are being followed on the computer are scheduled. This plan for slow integration of the system into the clinic has made ONCOCIN's initial release less disruptive to the clinic routine than it would have been if we had attempted to incorporate a comprehensive system that handled all patients and protocols from the onset. This method of integration has also allowed us to fine-tune our system early in its development, based on responses and suggestions from our physician users.

32.3.5 Responses and Modifications to the System

After the system's initial release, the data manager and the collaborating oncologists collected comments and suggestions from the physicians who used the system. We have made numerous program changes in response to suggestions for modifications and desirable new features. We have also conducted a number of formal studies to evaluate the impact of the system on physicians' attitudes, the completeness and accuracy of data collection, and the quality of the therapeutic decisions.

We soon learned that some of our initial design decisions had failed to anticipate important physician concerns. For example, if the Reasoner needed an answer to a special question not on the regular flow sheet form, our initial approach was to have the Interviewer interrupt data entry to request this additional information. The physicians were annoyed by these interruptions, so we modified the scheme to insert the question less obtrusively on a later section of the flow sheet, and to stop forcing the physician to answer such questions.

Another concern was that ONCOCIN was too stringent about its drug dosage recommendations, requesting justifications from the physician even for minor changes. We needed to take into account, for example, that a different pill size might decrease or increase a dose slightly and yet would be preferable for a patient's convenience. We subsequently obtained from the oncologists on our team ranges for each chemotherapeutic agent, within which any dosage modifications could be considered insignificant. Such minor modifications no longer generate requests for justification.[4] We also modified the program to recommend the same dose that the physician prescribed during a prior visit if that recommendation is within the acceptable range calculated by the program.

Some system users also asked whether the program could generate a progress note for the patient's visit. When we developed this feature and installed a small printer to prepare these notes in duplicate, use of the system was immediately made more desirable because this capability saved the physician the time required to dictate a note. This feature also helps to encourage the physician to enter relevant data completely and accurately because the quality of the resulting progress note is dependent on the data entry process.

When the system was first released, it was available only on the three mornings per week when the majority of lymphoma patients were seen (the computer, a DEC System 2020, is used at other times by other members of our research community). This allowed us to provide rapid response time through an arrangement for high-priority use of the computer. Since some lymphoma protocol patients were seen at other times, however, there were continuing problems in keeping the computer-based files up to date and thus in establishing ONCOCIN's role as a reliable aid for the management of that subset of patients. In response to this problem, we have made the system available whenever a patient known to the system is seen in the clinic. When the physician initiates a consultation, the program checks to see if the computer response is likely to be slow and, if so, prints out a warning to that effect. The physician may then either abort the session or proceed with the anticipation that the interaction will take longer than usual. We have found that the physicians understand and appreciate this feature and will often continue despite the delays.

[4]Current research is also investigating an adaptation of ONCOCIN's recommendation scheme whereby it will critique the physician's own therapy plan and give advice only when specifically requested to do so (Langlotz and Shortliffe, 1983).

32.3.6 Lessons Learned

It is clear that in order for a computer-based consultant to be effective in a clinical setting, the overall system design must take into account both the needs of the intended users and the constraints under which they function. This is the central theme of the lessons that we have learned from the MYCIN and ONCOCIN experiences. The program must be designed to satisfy a need for consultation and to provide this assistance in a fast, easy-to-use, and tactful manner. It should ideally avoid an incremental time commitment or an increase in the responsibilities of its users, or they will tend to resist its use. We have found that providing extra information-processing services, such as printing progress notes for the physicians, significantly heightens the system's appeal.

For ONCOCIN to have an effective role as a physician's assistant, providing both data management functions and consultations on patient treatment, it needs to be part of the daily routine in the clinic. Because of the limited number of patients and protocols currently on the system, ONCOCIN is still an exception to the daily routine; this will change as more protocols are encoded and the system is transferred to dedicated hardware. We are planning to move ONCOCIN to a personal workstation (a LISP machine capable of handling large AI programs) so that it will be self-contained. As it becomes the principal record-keeping system in the oncology clinic and enables the oncologists to receive useful advice for essentially all of their patient encounters, ONCOCIN will become successfully integrated into the clinic setting. The next stage will be to disseminate the system, mounted on single-user workstations, into other settings outside Stanford.

Physician involvement in the design of ONCOCIN has been crucial in all aspects of the system development. The collaborating oncologists provide answers to questions that are unclear from the protocol descriptions, evaluate the program's recommendations to ensure they are reasonable, offer useful feedback during the development of the user interface, and provide advice about how the computer-based consultation system can best fit into the clinic setting. Their collaboration and that of the computer scientists, medical personnel, and others in our interdisciplinary group (all of whom are committed to the creation of a clinically useful consultation tool) have combined to create a system for which limited integration into a clinical setting has been accomplished. We expect that total integration will be feasible within the next few years.