Clinical Decision-Support Systems

MARK A. MUSEN, YUVAL SHAHAR, AND EDWARD H. SHORTLIFFE

After reading this chapter, you should know the answers to these questions:

- What are three requirements for an excellent decision-making system?
- What are three decision-support roles for computers in clinical medicine?
- How has the use of computers for clinical decision support evolved since the 1960s?
- What is a knowledge-based system?
- What influences account for the gradual improvement in professional attitudes toward use of computers for clinical decision support?
- What are the five dimensions that characterize clinical decision-support tools?
- What are clinical-practice guidelines, and what are the challenges in providing guideline-based decision support?
- What are the principal scientific challenges in building useful and acceptable clinical decision-support tools?
- What legal and regulatory barriers could affect distribution of clinical decision-support technologies?

20.1 The Nature of Clinical Decision-Making

If you ask people what the phrase “computers in medicine” means, they often describe a computer program that helps physicians to make diagnoses. Although computers play numerous important clinical roles, from the earliest days of computing people have recognized that computers might support health-care workers by helping these people to sift through the vast collection of possible diseases and symptoms. This idea has been echoed in futuristic works of science fiction. In Star Trek, for example, medical workers routinely point devices at injured crew members to determine instantly what is the problem and how serious is the damage. The prevalence of such expectations, coupled with a general societal concern about the influence of computers on interpersonal relationships and on job security, has naturally raised questions among health workers. Just what can computers do today to support clinical decision-making? How soon will diagnostic tools be generally available? How good will they be? What will their effects be on the practice of medicine, on medical education, and on relationships among colleagues or between physicians and patients?

We can view the contents of this entire book as addressing clinical data and decision-making. In Chapter 2, we discussed the central role of accurate, complete, and relevant data in supporting the decisions that confront clinicians and other healthcare workers. In
Chapter 3, we described the nature of good decisions and the need for clinicians to understand the proper use of information if they are to be effective and efficient decision-makers. In Chapter 4 we introduced the cognitive issues that underly clinical decision making and that influence the design of systems for decision support. Subsequent chapters have mentioned many real or potential uses of computers to assist with such decision-making. Medical practice is medical decision-making, so most applications of computers in health care are intended to have a direct or tangential effect on the quality of healthcare decisions. In this chapter, we bring together these themes by concentrating on systems that have been developed specifically to assist health workers in making decisions.

20.1.1 Types of Decisions

By now, you are familiar with the range of clinical decisions. The classic problem of diagnosis (analyzing available data to determine the pathophysiologic explanation for a patient’s symptoms) is only one of these. Equally challenging, as emphasized in Chapters 3 and 4, is the diagnostic process—deciding which questions to ask, tests to order, or procedures to perform and determining the value of the results relative to associated risks or financial costs. Thus, diagnosis involves not only deciding what is true about a patient but also what data are needed to determine what is true. Even when the diagnosis is known, there often are challenging management decisions that test the physician’s knowledge and experience: Should I treat the patient or allow the process to resolve on its own? If treatment is indicated, what should it be? How should I use the patient’s response to therapy to guide me in determining whether an alternate approach should be tried or, in some cases, to question whether my initial diagnosis was incorrect after all?

Biomedicine is also replete with decision tasks that do not involve specific patients or their diseases. Consider, for example, the biomedical scientist who is using laboratory data to help with the design of her next experiment or the hospital administrator who uses management data to guide decisions about resource allocation in his hospital. Although we focus on systems to assist with clinical decisions in this chapter, we emphasize that the concepts discussed generalize to many other problem areas as well. In Chapter 23, for example, we examine the need for formal decision techniques and tools in creating health policies. The requirements for excellent decision-making fall into three principal categories: (1) accurate data, (2) pertinent knowledge, and (3) appropriate problem-solving skills.

The data about a case must be adequate for making an informed decision, but they must not be excessive. Indeed, a major challenge occurs when decision-makers are bombarded with so much information that they cannot process and synthesize the information intelligently and rapidly (see, for example, Chapter 17). Thus, it is important to know when additional data will confuse rather than clarify and when it is imperative to use tools (computational or otherwise) that permit data to be summarized for easier cognitive management (see Chapter 4). The operating room and intensive-care units are classic settings for this problem; patients are monitored extensively, numerous data are collected, and decisions often have to be made on an emergent basis. Equally important
is the quality of the available data. In Chapter 2, we discussed imprecision in terminology, illegibility and inaccessibility of records, and other opportunities for misinterpretation of data. Similarly, measurement instruments or recorded data may simply be erroneous; use of faulty data can have serious adverse effects on patient-care decisions. Thus, clinical data often need to be validated. Even good data are useless if we do not have the basic knowledge necessary to apply them properly. Decision-makers must have broad knowledge of medicine, in-depth familiarity with their area of expertise, and access to information resources that provide pertinent additional information. Their knowledge must be accurate, with areas of controversy well understood and questions of personal choice well distinguished from topics where a dogmatic approach is appropriate. Their knowledge must also be current; in the rapidly changing world of medicine, facts decay just as certainly as dead tissue does.

Good data and an extensive factual knowledge base still do not guarantee a good decision; good problem-solving skills are equally important. Decision-makers must know how to set appropriate goals for a task, how to reason about each goal, and how to make explicit the trade-offs between costs and benefits of diagnostic procedures or therapeutic maneuvers. The skilled clinician draws extensively on personal experience, and new physicians soon realize that good clinical judgment is based as much on an ability to reason effectively and appropriately about what to do as it is on formal knowledge of the field or access to high-quality patient data. Thus, clinicians must develop a strategic approach to test selection and interpretation, understand ideas of sensitivity and specificity, and be able to assess the urgency of a situation. Awareness of biases (see Chapter 3) and of the ways that they can creep into problem-solving also are crucial. This brief review of issues central to clinical decision-making serves as a fitting introduction to the topic of computer-assisted decision-making: Precisely the same topics are pertinent when we develop a computational tool for clinical problem-solving. The programs must have access to good data, they must have extensive background knowledge encoded for the clinical domain in question, and they must embody an intelligent approach to problem-solving that is sensitive to requirements for proper analysis, appropriate cost–benefit trade-offs, and efficiency.

### 20.1.2 The Role of Computers in Decision Support

A **clinical decision-support system** is any computer program designed to help healthcare professionals to make clinical decisions. In a sense, any computer system that deals with clinical data or knowledge is intended to provide decision support. It is accordingly useful to consider three types of decision-support functions, ranging from generalized to patient specific.

#### Tools for Information Management

Health-care information systems (Chapter 13) and information-retrieval systems (Chapter 19) are tools that manage information. Specialized knowledge-management workstations are under development in research settings; these workstations provide sophisticated environments for storing and retrieving clinical knowledge, browsing
through that knowledge much as we might page through a textbook, and augmenting it
with personal notes and information that we may need later for clinical problem-
solving. Information-management tools provide the data and knowledge needed by the
clinician, but they generally do not help her to apply that information to a particular
decision task. Interpretation is left to the clinician, as is the decision about what inform-
ation is needed to resolve the clinical problem.

Tools for Focusing Attention

Clinical-laboratory systems that flag abnormal values or that provide lists of possible
explanations for those abnormalities and pharmacy systems that alert providers to pos-
sible drug interactions (Evans et al., 1986; Tatro et al., 1975) are tools that focus the
user’s attention. Such programs are designed to remind the user of diagnoses or prob-
lems that might otherwise have been overlooked. Typically, they use simple logics, dis-
playing fixed lists or paragraphs as a standard response to a definite or potential
abnormality.

Tools for Providing Patient-Specific Recommendations

Such programs provide custom-tailored assessments or advice based on sets of patient-
specific data. They may follow simple logics (such as algorithms), may be based on deci-
sion theory and cost–benefit analysis, or may use numerical approaches only as an
adjunct to symbolic problem solving. Some diagnostic assistants (such as DXplain
[Barnett et al., 1987] or QMR [Miller et al., 1986]) suggest differential diagnoses or indi-
cate additional information that would help to narrow the range of etiologic possibili-
ties. Other systems (such as the original Internist-I program [Miller et al., 1982], from
which QMR was derived) suggest a single best explanation for a patient’s symptomat-
ology. Other systems interpret and summarize the patient’s record over time in a man-
ner sensitive to the clinical context (Shahar & Musen, 1996). Still other systems provide
therapy advice rather than diagnostic assistance (Musen et al., 1996).

The boundaries among these three categories are not crisp, but the distinctions are
useful in defining the range of capabilities that computers can provide to assist cli-
nicians with making decisions. Systems of the first two types are discussed elsewhere in
this book. For example, Chapters 12 through 18 describe systems that contain and
manipulate patient data that are of importance in reaching good clinical decisions.
Chapters 19 and 21 discuss methods for accessing information, knowledge, and the
accumulated experience of other professionals. In this chapter, we focus on the third cat-
egory: patient-specific systems.

20.2 Historical Perspective

Since the earliest days of computers, health professionals have anticipated the time when
machines would assist them in the diagnostic process. The first articles dealing with this
possibility appeared in the late 1950s (Ledley & Lusted, 1959), and experimental
prototypes appeared within a few years (Warner et al., 1964). Many problems prevented
the widespread introduction of such systems, however, ranging from the limitations of
the scientific underpinnings to the logistical difficulties that developers encountered
when encouraging clinicians to use and accept systems that were not well integrated into
the practitioners’ usual workflow.

Three advisory systems from the 1970s provide a useful overview of the origin of
work on clinical decision-support systems: deDombal’s system for diagnosis of abdomi-
nal pain (de Dombal et al., 1972), Shortliffe’s MYCIN system for selection of antibi-
otic therapy (Shortliffe, 1976), and the HELP system for delivery of inpatient medical
alerts (Kuperman et al., 1991; Warner, 1979).

20.2.1 Leeds Abdominal Pain System

Starting in the late 1960s, F. T. deDombal and his associates at the University of Leeds
studied the diagnostic process and developed computer-based decision aids using
Bayesian probability theory (see Chapter 3). Using surgical or pathologic diagnoses as
the gold standard, they emphasized the importance of deriving the conditional proba-
bilities used in Bayesian reasoning from high-quality data that they gathered by collect-
ing information on thousands of patients (Adams et al., 1986). Their system, the Leeds
abdominal pain system, used sensitivity, specificity, and disease-prevalence data for vari-
cous signs, symptoms, and test results to calculate, using Bayes’ theorem, the probabil-
ity of seven possible explanations for acute abdominal pain (appendicitis, diverticulitis,
perforated ulcer, cholecystitis, small-bowel obstruction, pancreatitis, and nonspecific
abdominal pain). To keep the Bayesian computations manageable, the program made
the assumptions of (1) conditional independence of the findings for the various diag-
noses and (2) mutual exclusivity of the seven diagnoses (see Chapter 3).

In one system evaluation (de Dombal et al., 1972), physicians filled out data sheets
summarizing clinical and laboratory findings for 304 patients who came to the emer-
gency room with abdominal pain of sudden onset. The data from these sheets became
the attributes that were analyzed using Bayes’ rule. Thus, the Bayesian formulation
assumed that each patient had one of the seven conditions and selected the most likely
one on the basis of the recorded observations. Had the program been used directly by
emergency-room physicians, results could have been available, on average, within 5 min-
utes after the data form was completed. During the study, however, the cases were run
in batch mode; the computer-generated diagnoses were saved for later comparison to (1)
the diagnoses reached by the attending clinicians and (2) the ultimate diagnosis verified
during surgery or through appropriate tests.

In contrast to the clinicians’ diagnoses, which were correct in only 65 to 80 percent
of the 304 cases (with accuracy depending on the individual clinician’s training and expe-
rience), the program’s diagnoses were correct in 91.8 percent of cases. Furthermore, in
six of the seven disease categories, the computer was more likely to assign the patients
to the correct disease category than was the senior clinician in charge of the case. Of
particular interest was the program’s accuracy regarding appendicitis—a diagnosis that
is often made incorrectly (or, less often, is missed or at least delayed). In no cases of
appendicitis did the computer fail to make the correct diagnosis, and in only six cases
were patients with nonspecific abdominal pain incorrectly classified as having appendicitis. Based on the actual clinical decisions, however, more than 20 patients with nonspecific abdominal pain underwent unnecessary surgery for an incorrect diagnosis of appendicitis, and six patients who did have appendicitis were observed for more than 8 hours before they were finally taken to the operating room.

With the introduction of personal computers, deDombal’s system began to achieve widespread use—from emergency departments in other countries to the British submarine fleet. Surprisingly, the system has never obtained the same degree of diagnostic accuracy in other settings that it did in Leeds—even when adjustments were made for differences in prior probabilities of disease. There are several reasons possible for this discrepancy. The most likely explanation is that there may be considerable variation in the way that clinicians interpret the data that must be entered into the computer. For example, physicians with different training or from different cultures may not agree on the criteria for identification of certain patient findings on physical examination, such as “rebound tenderness.” Another possible explanation is that there are different probabilistic relationships between findings and diagnoses in different patient populations.

20.2.2 MYCIN

A different approach to computer-assisted decision support was embodied in the MYCIN program, a consultation system that de-emphasized diagnosis to concentrate on appropriate management of patients who have infections (Shortliffe, 1976). MYCIN’s developers believed that straightforward algorithms or statistical approaches were inadequate for this clinical problem in which the nature of expertise was poorly understood and even the experts often disagreed about how best to manage specific patients, especially before definitive culture results became available. As a result, the researchers were drawn to the field of artificial intelligence (AI), a subfield of computer science that has focused on manipulation of abstract symbols rather than on numerical calculations.

Knowledge of infectious diseases in MYCIN was represented as production rules, each containing a “packet” of knowledge derived from discussions with collaborating experts (Figure 20.1). A production rule is simply a conditional statement that relates observations to associated inferences that can be drawn. MYCIN’s power was derived from such rules in a variety of ways:

- The MYCIN program determined which rules to use and how to chain them together to make decisions about a specific case.
- The rules often formed a coherent explanation of MYCIN’s reasoning—those that applied to the current decision were displayed in response to users’ questions (Figure 20.2). Although rules were stored in a machine-readable format, English translations could be displayed.
- By removing, altering, or adding rules, system developers could modify the program’s knowledge structures rapidly, without explicitly reprogramming or restructuring other parts of the knowledge base. (Making such changes, however, could lead to unintended side effects.)
Figure 20.1. A typical rule from the MYCIN system. Rules are conditional statements that indicate what conclusions can be reached or actions taken if a specified set of conditions is found to be true. In this rule, MYCIN is able to conclude probable bacterial causes of infection if the five conditions in the premise are all found to be true for a specific patient. Not shown are the measures of uncertainty that are also associated with inference in the MYCIN system.

\[
\text{Rule507} \\
\text{IF:} \\
1) \text{The infection that requires therapy is meningitis,} \\
2) \text{Organisms were not seen on the stain of the culture,} \\
3) \text{The type of infection is bacterial,} \\
4) \text{The patient does not have a head injury defect, and} \\
5) \text{The age of the patient is between 15 years and 55 years} \\
\text{THEN:} \\
\text{The organisms that might be causing the infection are diplococcus-pneumoniae and neisseria-meningitis}
\]

**Figure 20.2.** Two examples of MYCIN’s explanation capabilities. User input is shown in boldface capital letters and follows the double asterisks. The system expands each ["WHY"] question (enclosed in square brackets) to ensure that the user is aware of its interpretation of the query.
The developers evaluated MYCIN’s performance on therapy selection for patients with blood-borne bacterial infections (Yu 1979b), and for those with meningitis (Yu et al., 1979a). In the latter study, MYCIN gave advice that compared favorably with that offered by experts in infectious diseases. MYCIN, however, is best viewed as an early exploration of methods for capturing and applying ill-structured expert knowledge to solve important medical problems. Although the program was never used clinically, it paved the way for a great deal of research and development in the 1980s. In fact, the development of knowledge-based systems, and the commercialization of the rule-based approach in a variety of nonmedical fields during the early 1980s, evolved from MYCIN and from related systems developed during the 1970s (Hayes-Roth et al., 1983).

20.2.3 HELP

In the earlier discussion of computer-based patient record systems (Chapter 12), we referred to the HELP system, an integrated hospital information system developed at LDS Hospital in Salt Lake City. HELP has the ability to generate alerts when abnormalities in the patient record are noted, and its impact on the development of the field has been immense, with applications and methodologies that span nearly the full range of activities in biomedical informatics (Kuperman et al., 1991).

HELP adds to a conventional medical-record system a monitoring program and a mechanism for storing decision logic in “HELP sectors” or logic modules. Thus, patient data are available to users who wish to request specific information, and the usual reports and schedules are automatically printed or otherwise communicated by the system. In addition, there is a mechanism for event-driven generation of specialized warnings, alerts, and reports. HELP’s developers originally created a specialized language named PAL for writing medical knowledge in HELP sectors. Beginning in the 1990s, workers at LDS Hospital, Columbia Presbyterian Medical Center, and elsewhere created and adopted a standard formalism for encoding decision rules known as the Arden syntax—a programming language that provides a canonical means for writing rules that relate specific patient situations to appropriate actions for practitioners to follow (Hripcsak et al., 1994). The Arden syntax incorporates many of the features of PAL, as well as those of other frameworks for writing clinical decision rules that other research groups developed during the 1970s and 1980s. In the Arden syntax, each decision rule, or HELP sector, is called a medical logic module (MLM). Figure 20.3 shows one such MLM and its representation in the Arden syntax.

Whenever new data about a patient become available, regardless of the source, the HELP system checks to see whether the data match the criteria for invoking an MLM. If they do, the system evaluates the MLM to see whether that MLM is relevant for the specific patient. The logic in these MLMs has been developed by clinical experts working with medical information scientists. The output generated by successful MLMs includes, for example, alerts regarding untoward drug actions, interpretations of laboratory tests, or calculations of the likelihood of diseases. This output result is communicated to the appropriate people through the hospital information system’s workstations or on written reports, depending on the urgency of the output message and the location and functions of the person for whom the report is intended.
From the 1970s to the beginning of the current decade, HELP served as a superb example of how the integration of decision support with other system functions can heighten a program’s acceptance and encourage its use. Several studies (e.g., Evans et al., 1986) demonstrated the beneficial effect of HELP’s decision logic on clinical measurements at LDS Hospital. Alerts and warnings were produced through the normal collection of patient data; transcription of data for reuse in secondary settings were avoided through the full integration of the computing environment. As discussed in Chapter 13, hospital systems have evolved toward more distributed architectures, with desktop or handheld computers serving as workstations and data being shared over local-area networks, sometimes with wireless connectivity. This large project at LDS Hospital has served as an important model of how decision support through integrated

penicillin_order :=
  event {medication_order
    where class = penicillin};

/* find allergies */
penicillin_allergy :=
  read last {allergy
    where agent_class = penicillin};

;*
  evoke: penicillin_order ;*
logic:
If exist (penicillin_allergy) then conclude true;
endif;

;*
  action:
write
"Caution, the patient has the following allergy to penicillin documented:" || penicillin_allergy ;*

Figure 20.3. This medical logic module (MLM), written in the Arden syntax, prints a warning for healthcare workers whenever a patient who reportedly is allergic to penicillin receives a prescription for a drug in the penicillin class. The evoke slot defines a situation that causes the rule to be triggered; the logic slot encodes the decision logic of the rule; the action slot defines the procedure to follow if the logic slot reaches a positive conclusion.

From the 1970s to the beginning of the current decade, HELP served as a superb example of how the integration of decision support with other system functions can heighten a program’s acceptance and encourage its use. Several studies (e.g., Evans et al., 1986) demonstrated the beneficial effect of HELP’s decision logic on clinical measurements at LDS Hospital. Alerts and warnings were produced through the normal collection of patient data; transcription of data for reuse in secondary settings were avoided through the full integration of the computing environment. As discussed in Chapter 13, hospital systems have evolved toward more distributed architectures, with desktop or handheld computers serving as workstations and data being shared over local-area networks, sometimes with wireless connectivity. This large project at LDS Hospital has served as an important model of how decision support through integrated
data monitoring can bypass many of the traditional barriers to the use of computers for clinical decision support. Ideas from HELP are being incorporated into decision-support components from several commercial vendors of clinical information systems and EHRs.

### 20.2.4 Lessons from Early Decision-Support Systems

The Leeds abdominal pain system was an important exemplar of the clinical value of Bayesian diagnostic systems. Subsequent Bayesian systems, such as the Pathfinder system for diagnosis of lymph-node pathology (Heckerman et al., 1989), built solidly on the foundation laid by deDombal and his co-workers. Similarly, rule-based approaches to clinical decision-making, as pioneered in systems such as MYCIN and HELP, have led to more recent frameworks for representing medical knowledge, such as the Arden syntax. The early decision-support systems demonstrated the feasibility of encoding medical knowledge so that it could be processed by computers. They also have helped researchers in biomedical informatics to clarify both the strengths and limitations of alternative knowledge-representation approaches.

Although the HELP system was a notable exception, most early decision-support tools were rarely used by health personnel and were viewed with skepticism. The subsequent evolution in attitudes has been due in large part to four influences: (1) the emergence of personal workstations, the World Wide Web, and easy-to-use interfaces; (2) the increasing recognition on the part of technology developers that computer systems must meld transparently with the work practices of groups that are asked to adopt new technologies; (3) the growing distress among health professionals and managed-care organizations regarding the amount of information that practitioners need to practice medicine well and to avoid errors; and (4) the increasing fiscal pressure to practice cost-effective, evidence-based medicine, which leads practitioners to consider carefully the clinical utility and reliability of tests, procedures, and therapies—especially when the latter are expensive or risky.

Gradual changes in attitudes and increasing acceptance of the ideas of computer-based decision tools for healthcare professionals are of course not in themselves adequate to ensure developmental progress and the adoption of new information-management facilities. Current enthusiasm will sour rapidly if the products of research are not responsive to real-world needs and are not sensitive to the logistical requirements of the practice settings in which clinicians work.

### 20.3 A Structure for Characterizing Clinical Decision-Support Systems

If we are to assess adequately any new clinical decision-support tool or to understand the range of issues that can affect the chances for successful implementation, we must have an organizing framework for considering such programs. One approach is to characterize decision-support systems along five dimensions: (1) the system’s intended function, (2) the mode by which advice is offered, (3) the consultation style, (4) the
underlying decision-making process, and (5) the factors related to human-computer interaction. As this spectrum of considerations suggests, excellent decision-making capabilities alone do not guarantee system utility or acceptance.

20.3.1 System Function

Decision-support programs generally fall into two categories: those that assist healthcare workers with determining what is true about a patient (usually what the correct diagnosis is—as in the Leeds abdominal-pain system) and those that assist with decisions about what to do for the patient (usually what test to order, whether to treat, or what therapy plan to institute—as in MYCIN). Many systems assist clinicians with both activities (e.g., diagnostic programs often help physicians to decide what additional information would be most useful in narrowing the differential diagnosis for a given case), but the distinction is important because advice about what to do for a patient cannot be formulated without balancing the costs and benefits of action. Determination of what is true about a patient, based on a fixed set of data that are already available, can theoretically be made without consideration of cost and risk. Thus, a “pure” diagnostic program leaves to the user the task of deciding what data to gather or requires a fixed set of data for all patients. As all practitioners know, however, it is unrealistic to view making a diagnosis as separable from the process of choosing from the available options for data collection and therapy. Moreover, many physicians believe that the majority of questions about which they seek consultation deal with what they should do rather than with what is true about a patient given a fixed data set.

20.3.2 The Mode for Giving Advice

Like the abdominal pain program and MYCIN, most decision-support programs have assumed a passive role in giving advice to clinicians (Reggia & Turhim, 1985). Under this model, the practitioner must recognize when advice would be useful and then must make an explicit effort to access the computer program; the decision-support system waits for the user to come to it. The clinician then describes a case by entering data and requests a diagnostic or therapeutic assessment.

There are also technologies, such as the HELP system, that play a more active role, providing decision support as a byproduct of monitoring or of data-management activities; such systems do not wait for physicians or other health workers specifically to ask for assistance. A great appeal of such systems is their ability to give assistance to healthcare workers without requiring laborious data entry by the clinicians themselves. Such capabilities are possible only because the system’s decision logic is integrated with a comprehensive database of patient information that is already being gathered from diverse sources within the healthcare enterprise. Because practitioners generally do not request assistance from such systems, but instead receive it whenever monitored patient data warrant it, one challenge is to avoid generating excessive numbers of warnings for minor problems already likely to be understood. Otherwise, such “false-positive” advisory reports can generate antagonistic responses from users and can blunt the usefulness of those warnings that have greater clinical significance.
20.3.3 Style of Communication

Decision-support systems have tended to operate under one of two styles of interaction: the consulting model or the critiquing model. In the consulting model, the program serves as an advisor, accepting patient-specific data, possibly asking questions, and generating advice for the user about diagnosis or management. For example, MYCIN was an early example of a program that adopted the consulting approach. In the critiquing model, on the other hand, the clinician has a preconceived idea of what is happening with a patient or what management plan would be appropriate. The computer then acts as a sounding board for the user's own ideas, expressing agreement or suggesting reasoned alternatives. A pioneering example of a critiquing system was ATTENDING, a standalone program that critiqued a patient-specific plan for anesthetic selection, induction, and administration after that plan had been proposed by the anesthesiologist who would be managing the case (Miller, 1986). Such critiquing systems meet many physicians' desires to formulate plans on their own but to have those plans double-checked occasionally before acting on them. In the critiquing style, the program focuses more directly on the plan in which the physician is interested.

The critiquing model also can be applied in an active monitoring setting. For example, the HELP system monitored physicians' drug-therapy decisions and suggested alternate approaches that might be preferable (Evans et al., 1986). Similarly, the HyperCritic system (van der Lei & Musen, 1991) offered suggestions regarding how primary-care physicians might improve their management of patients with hypertension by performing a behind-the-scenes analysis of the patients' computer-based record at the time of each clinic visit.

20.3.4 Underlying Decision-Making Process

A wide variety of techniques has been used in the design and implementation of decision-support systems. The simplest logics have involved problem-specific flowcharts designed by clinicians and then encoded for use by a computer. Although such algorithms have been useful for triage purposes and as a didactic technique used in journals and books where an overview for a problem's management has been appropriate, they have been largely rejected by physicians as too simplistic for routine use (Grimm et al., 1975). In addition, the advantage of their implementation on computers has not been clear; the use of simple printed copies of the algorithms generally has proved adequate (Komaroff et al., 1974). A noteworthy exception is a large computer program first described in the early 1970s at the Beth Israel Hospital in Boston (Bleich, 1972); it used a detailed algorithmic logic to provide advice regarding the diagnosis and management of acid–base and electrolyte disorders. Although additional techniques—such as mathematical modeling, pattern recognition, and statistical analysis of large databases—have been used in experimental decision-support systems (Shortliffe et al., 1979), the predominant methods have been drawn from Bayesian modeling, decision analysis, artificial neural networks, and AI.

Because computers were traditionally viewed as numerical calculating machines, people had recognized by the 1960s that they could be used to compute the pertinent
probabilities based on observations of patient-specific parameters (as long as each had a known statistical relationship to the possible disease etiologies). Large numbers of Bayesian diagnosis programs have been developed in the intervening years, many of which have been shown to be accurate in selecting among competing explanations of a patient’s disease state (Heckerman & Nathwani, 1992). As we mentioned earlier, among the largest experiments have been those of deDombal and associates (1972) in England, who adopted a simple Bayesian model that assumed that there are no conditional dependencies among findings (e.g., the fallacious assumption that the presence of a finding such as fever never affects the likelihood of the presence of a finding such as chills). More recent work on the use of belief networks for automated decision-making has demonstrated that it is practical to develop more expressive Bayesian systems in which conditional dependencies can be modeled explicitly rather than ignored. (Belief networks are described in Chapter 3.)

Because making most decisions in medicine requires weighing the costs and benefits of actions that could be taken in diagnosing or managing a patient’s illness, researchers also have developed tools that draw on the methods of decision analysis (Sax et al., 1988; Weinstein & Fineberg, 1980). Decision analysis adds to Bayesian reasoning the idea of explicit decisions and of utilities associated with the various outcomes that could occur in response to those decisions (see Chapter 3). One class of programs is designed for use by the analysts themselves; such programs assume a detailed knowledge of decision analysis and would be of little use to the average clinician (Pauker & Kassirer, 1981). A second class of programs uses decision-analysis concepts within systems designed to advise physicians who are not trained in these techniques. In such programs, the underlying decision models generally have been prespecified—either as decision trees that enumerate all possible decisions and all possible ramifications of those decisions or as belief networks in which explicit decision and utility nodes are added, called influence diagrams.

There has been considerable interest in the use of artificial neural networks as the basis for automated medical diagnosis. Artificial neural networks (ANNs) are computer programs that perform classification, taking as input a set of findings that describe a given case and generating as output a set of numbers, where each output corresponds to the likelihood of a particular classification that could explain the findings. The program performs this function by propagating carefully calculated weights through a network of several layers of nodes. The structure of the network is uniform for any class of decision problem; the weights associated with each of the nodes, however, are tuned so that the network tends to generate the correct classification for any set of inputs. The values for the weights are determined in incremental fashion when a network is trained on a large collection of previously classified examples during a period of supervised learning. Like statistical pattern-recognition methods, artificial neural networks translate a set of findings into a set of weighted classifications consistent with those findings. Unfortunately, there is no way that an observer can directly understand why an artificial neural network might reach a particular conclusion. Artificial neural networks may have significant advantages, however, when the correct diagnosis may depend on interactions among the findings that are difficult to predict.
Since the early 1970s, a growing body of researchers have been applying AI techniques to the development of diagnostic and therapy-management systems (Clancey, 1984; Miller, 1988; Szolovits, 1982). We have already discussed the MYCIN system, an important early example of work in this area. Artificial intelligence traditionally has been closely tied to psychology and to the modeling of logical processes by computer (see Chapter 4). Psychological studies of how medical experts perform problem-solving (Elstein et al., 1978; Kupiers & Kassirer, 1984) therefore have been influential in much research in medical AI. Of particular pertinence to the development of decision-support systems is the subfield of AI research that is concerned with knowledge-based systems. A knowledge-based system is a program that symbolically encodes concepts derived from experts in a field—in a knowledge base—and that uses that knowledge base to provide the kind of problem analysis and advice that the expert might provide.

Clinical decision-making often requires reasoning under uncertainty. Knowledge-based systems in medicine have consequently incorporated either Bayesian or ad hoc schemes for dealing with partial evidence and with uncertainty regarding the effects of proposed interventions. What is most characteristic of a knowledge-based system, however, is that the knowledge base encodes a non-numeric, qualitative model of how inferences are related to reach abstract conclusions about a case (e.g., the diseases that a patient might have, the therapy that should be administered, the laboratory tests that should be ordered) (Clancey, 1989). Thus, instead of modeling the relationships among patient findings and possible diagnoses purely in terms of statistical associations or mathematical equations, knowledge-based systems might represent those relationships in terms of qualitative, symbolic structures. Production rules such as those in MYCIN (see Fig. 20.1) often have been used to build knowledge-based systems, as have many other approaches (David et al., 1993). The knowledge in a knowledge-based system may include probabilistic relations, such as between symptoms and underlying diseases. Typically, such relations are augmented by qualitative relations, such as causality and temporal relations.

20.3.5 Human–Computer Interaction

There is perhaps no omission that, historically, accounts more fully for the impracticality of many clinical decision tools than the failure of developers to deal adequately with the logistical, mechanical, and psychological aspects of system use (see also Chapter 4). Often, system builders have concentrated primarily on creating computer programs that can reach good decisions. Yet researchers have shown repeatedly that an ability to make correct diagnoses, or to suggest therapy similar to that recommended by human consultants, is only one part of the formula for system success. Fortunately, there is increasing recognition that decision-support systems should, at the very least, present interfaces to their users that are uncluttered and intuitive, where users can predict in advance the consequences of their actions (and undo those actions, if necessary). At best, the decision-support element should be embedded within some larger computer system that is already part of the users’ professional routine—thus making decision support a byproduct of the practitioners’ ordinary work practices.
Many potential users of clinical decision-support tools have found their early enthusiasm dampened by programs that are cumbersome to access, slow to perform, and difficult to learn to use. Systems can fail, for example, if they require that a practitioner interrupt the normal pattern of patient care to move to a separate workstation or to follow complex, time-consuming startup procedures. Lengthy interactions, or ones that fail to convey the logic of what is happening on the screen, also discourage use of the program. Health professionals are likely to be particularly frustrated if the decision tool requires the manual reentry of information available on other computers. Solutions to such problems require sensitivity during the design process and, frequently, resolution of inadequacies at the institutional level. For example, linking computers to one another so that they can share data requires implementation of an overall networking and data-sharing strategy for the hospital or clinic. The advent of wireless networks that allow users to roam about a hospital or clinic, writing directly onto a computer tablet with a pen-based interface, offers solutions for both clinical computing in general and for access to decision-support systems in particular. Similarly, novel human–computer interfaces based on speech, gestures, and virtual reality offer new dimensions to the ways in which healthcare workers can interact with decision-support systems (see Chapter 24).

20.4 Construction of Decision-Support Tools

Despite significant research progress since the idea of computer-based medical decision-support systems first emerged, several barriers continue to impede the effective implementation of such tools in clinical settings. As we implied earlier, these obstacles include unresolved questions of both science and logistics.

20.4.1 Acquisition and Validation of Patient Data

As emphasized in Chapter 2, few problems are more challenging than the development of effective techniques for capturing patient data accurately, completely, and efficiently. You have read in this book about a wide variety of techniques for data entry, ranging from keyboard entry, to speech input, to methods that separate the clinician from the computer (such as scannable forms, real-time data monitoring, and intermediaries who transcribe written or dictated data for use by computers). All these methods have limitations, and healthcare workers frequently state that their use of computers will be limited unless they are freed of the task of data entry and can concentrate instead on data review and information retrieval (Shortliffe, 1989). Even if computers could accept unrestricted speech input, there would be serious challenges associated with properly structuring and encoding what was said. Otherwise, spoken input becomes a large free-text database that defies semantic interpretation. Many workers believe that some combination of speech and graphics, coupled with integrated data-management environments that will prevent the need for redundant entry of the same information into multiple computer systems within a hospital or clinic, are the key advances that will attract busy clinicians and other health workers to use computer-based tools.
The problems of data acquisition go beyond entry of the data themselves, however. A primary obstacle is that we lack standardized ways of expressing most clinical situations in a form that computers can interpret. There are several controlled medical terminologies that healthcare workers use to specify precise diagnostic evaluations (e.g., the International Classification of Diseases and SNOMED-CT), clinical procedures (e.g., Current Procedural Terminology), and so on (see Chapter 7). Still, there is no controlled terminology that captures the nuances of a patient’s history of present illness or findings on physical examination. There is no coding system that can reflect all the details of physicians’ or nurses’ progress notes. Given that much of the information in the medical record that we would like to use to drive decision support is not available in a structured, machine-understandable form, there are clear limitations on the data that can be used to assist clinician decision-making. Nevertheless, even when computer-based patient records store substantial information only as free-text entries, those data that are available in coded form (typically, diagnosis codes and prescription data) can be used to significant advantage (van der Lei et al., 1991). Finally, even full electronic medical records may not include all of the relevant patient-specific data (e.g., professional and marital problems) and thus should be viewed realistically as an incomplete source of information.

### 20.4.2 Modeling of Medical Knowledge

People who have attempted to acquire the knowledge for a medical decision-support system by reading a textbook or several journal articles and by trying to encode the implied knowledge in some program can attest to the complexity of translating from the usual text approach for communicating knowledge to a structure appropriate for the logical application of that knowledge by a computer. The problem is not unlike that of identifying what you as a reader need to do to interpret, internalize, and apply properly the wealth of information in a book such as this. Creation of a computer-based decision-support system thus requires substantial **modeling** activity: deciding what clinical distinctions and patient data are relevant, identifying the concepts and relationships among concepts that bear on the decision-making task, and ascertaining a problem-solving strategy that can use the relevant clinical knowledge to reach appropriate conclusions.

You cannot glean any of this information simply by reading a textbook; clinical experts themselves may not be able to verbalize the knowledge needed to solve even routine cases (Johnson, 1983). Consequently, construction of any decision-support system—regardless of the underlying decision-making methodology—entails development of a model of both the required problem-solving behavior and the clinical knowledge that will inform that problem-solving. Considerable work in biomedical informatics currently concentrates on the design of frameworks that allow system builders to model the knowledge that ultimately will be captured within decision-support tools. Abstract modeling methodologies such as Common KADS (Schreiber et al., 2000) have been widely adopted by commercial developers of decision-support systems, particularly in Europe. Development of computer-based tools that can assist in the modeling of clinical knowledge remains an active area of investigation (Eriksson et al., 1995; Musen et al., 1995; van Heijst et al., 1995).
20.4.3 Elicitation of Medical Knowledge

Researchers are devising methods that will facilitate the development and maintenance of medical knowledge bases (Musen, 1993). The rapid evolution of medical knowledge makes knowledge-base maintenance a particularly important problem. Investigators have developed a variety of computer programs that acquire the knowledge base for a decision-support program by interacting directly with the expert, the goal being to avoid the need for a computer programmer to serve as intermediary (Eriksson & Musen, 1993; Lanzola et al., 1995; Musen et al., 1987). In all these approaches, analysts must first work with clinical experts to model the relevant application area.

For example, early researchers used a special-purpose tool known as OPAL (Musen et al., 1987) (Fig. 20.4) to enter and maintain the knowledge base of the cancer-chemotherapy advisor ONCOCIN (Shortliffe, 1986); the developers of OPAL built into the tool a comprehensive model of cancer-chemotherapy administration, allowing OPAL to transform the process of knowledge elicitation for ONCOCIN into a matter of filling in the blanks of structured forms and of drawing flowchart diagrams on the computer screen. When creating domain-specific knowledge-elicitation tools such as OPAL, developers create their model of the intended application area for the target decision-support system and then either program that model by hand into the tool (as they did in the case of building the original OPAL program) or enter the model into a meta-tool (Eriksson & Musen, 1993), which then generates automatically a special-

![Diagram](image-url)

**Figure 20.4.** A clinical researcher can use OPAL to describe the overall schema of an ONCOCIN cancer-treatment plan using the graphical environment shown here. She creates the individual boxes by making selections from the palette of choices at the bottom of the screen and then positions and joins them as desired. The model of cancer chemotherapy built into OPAL determines that possible selections include chemotherapy (CHEMO in the diagram), X-ray therapy (XRT), as well as the idea of randomization and stratification of patients enrolled in clinical trials. The figure shows a relatively simple protocol in which patients are treated with a three-drug chemotherapy called VAM, followed by a four-drug chemotherapy called POCC, until there is complete response (CR).
purpose knowledge-elicitation tool based on that model. Protégé (see Section 20.5.2) is a meta-tool that many developers have used to create automatically domain-specific knowledge-elicitation tools like OPAL by taking as input analysts’ models of the relevant application areas (Musen, 1998; Musen et al., 1995).

20.4.4 Representation of and Reasoning About Medical Knowledge

Among the ongoing research challenges is the need to refine the computational techniques for encoding the wide range of knowledge used in problem-solving by medical experts. Although well-established techniques such as the use of frames or rules exist for storing factual or inferential knowledge, several complex challenges remain. For example, physicians use mental models of the three-dimensional relationships among body parts and organs when they are interpreting data or planning therapy. Representing such anatomical knowledge and performing spatial reasoning by computer have proved to be particularly challenging. Similarly, human beings have a remarkable ability to interpret changes in data over time, assessing temporal trends and developing models of disease progression or the response of disease to past therapies. Researchers continue to develop computer-based methods for modeling such tasks.

Another kind of expertise, often poorly recognized but clearly important to optimal knowledge management by computer-based tools, is the human skill inherent in knowing how to use what is known. In medicine, we often call this skill “good clinical judgment,” and we properly distinguish it from the memorization of factual knowledge or data from the literature. It is similarly clear that simply giving computers lots of factual knowledge will not make them skilled in a field unless they also are expert in the proper application of that knowledge. It is in this area particularly that improved understanding of the psychology of human problem-solving is helping researchers to develop decision-support tools that more closely simulate the process by which expert clinicians move from observations to diagnoses or management plans (see Chapter 4).

20.4.5 Validation of System Performance

Many observers are horrified when they imagine what they might have to do to validate and maintain the currency of large clinical knowledge bases. After all, medical knowledge is advancing at a rapid pace, and an advisory system that uses yesterday’s knowledge may fail to provide the best advice available for a patient’s problem. Although researchers with limited goals have been willing to take on responsibility for short-term knowledge-base maintenance in support of their scholarly activities, it is likely that professional organizations or other national bodies will in time need to assume responsibility for the currency and integrity of large clinical knowledge bases.

When a knowledge base is well validated, developers still face challenges in determining how best to evaluate the performance of the decision-support tools that use the knowledge. When a gold standard of performance exists, formal studies can
compare the program’s advice with that accepted standard of “correctness.” This tech-
nique is especially pertinent for diagnostic tools, where biopsy, surgery, or autopsy
data can be used as an appropriate gold standard. In the case of therapy-advice sys-
tems, however, the gold standard is more difficult to define. Even experts may disagree
about the proper way to treat a specific patient, and there can seldom be a realistic
controlled trial that attempts to show which approach is right in any absolute sense.
For this reason, workers have experimented with techniques that compare the recom-
mendations of a therapy-management program with those of experts (see Chapter 11).
With proper controls, such studies can be useful, although they have shown that even
experts in a field generally do not receive perfect marks when assessed by their peers.
The problem of evaluation remains a ripe area for further research (Friedman &
Wyatt, 1997a).

20.4.6 Integration of Decision-Support Tools

As we have emphasized, the successful introduction of decision-support tools is
likely to be tied to these tools’ effective integration with routine clinical tasks. We
need more innovative research on how best to tie knowledge-based computer tools to
programs designed to store, manipulate, and retrieve patient-specific information.
We explained how the HELP system included decision-support functions that are
triggered to generate warnings or reports whenever an internally specified set of con-
ditions holds for a given patient (Section 20.2.3). As hospitals and clinics increasingly
use multiple small machines optimized for different tasks, however, the challenges of
integration are inherently tied to issues of networking and systems interfaces. It is in
the electronic linking of multiple machines with overlapping functions and data
needs that the potential of distributed but integrated patient data processing will be
realized.

20.5 Illustrative Examples of Clinical Decision-Support Systems

To illustrate the state of the art and the ways in which new technologies have affected
the evolution of decision-support tools, we shall discuss selected features of several well-
known decision-support systems in two major areas: diagnosis and patient manage-
ment. Quick Medical Reference (like its predecessor, Internist-1) supports diagnostic
problem solving in general internal medicine, while DXplain is a continuously evolving
Web-based diagnostic system. The EON system is a representative example for one of
the recent guideline-based decision-support systems, which provide therapeutic recom-
mendations for treatment in accordance with predefined protocols. The systems dis-
cussed here in some detail demonstrate widely differing architectures. Quick Medical
Reference is used primarily as a standalone system, DXplain is a self-contained sys-
tem, but currently accessed mostly over the World Wide Web, and EON comprises a
set of software components that are designed to be integrated within larger clinical
information systems.
20.5.1 Diagnosis: The Internist-1/QMR Project and the DXplain System

We will demonstrate the task of supporting clinical diagnosis using two well known but very different systems: Internist-1, which evolved into the QMR system, and the DXplain system, which is an important Web-based resource.

20.5.1.1 The Internist-1/QMR project

**Internist-1** was a large diagnostic program that was developed at the University of Pittsburgh School of Medicine in the 1970s (Miller et al., 1982; Pople, 1982). The Internist-1 program subsequently grew into a decision-support system known as **Quick Medical Reference** (QMR). QMR was marketed commercially for several years, and was used by a large community of practitioners and students. Although currently the system is not actively supported, it has been highly influential and the subject of considerable study by the medical-informatics community.

The goal of the original Internist-1 project was to model diagnosis in general internal medicine. Internist-1 contained knowledge of almost 600 diseases and of nearly 4,500 interrelated *findings*, or *disease manifestations* (signs, symptoms, and other patient characteristics). On average, each disease was associated with between 75 and 100 findings. The task of diagnosis would be straightforward if each disease were associated with a unique set of findings. Most findings, however, such as fever, are associated with multiple disease processes, often with varying levels of likelihood for each disease. Clinicians have long recognized that it is not feasible to perform simple pattern matching to make difficult diagnoses. On the other hand, it is impractical to estimate conditional probabilities (such as those used by the Leeds abdominal pain system) for all the diseases and findings in Internist-1’s knowledge base—particularly because many of the 600 disease syndromes are rare and thus are not well described in the clinical literature. For these reasons, the developers of Internist-1 chose to create an ad hoc scoring scheme to encode the relationships between specific findings and diseases.

To construct the Internist-1 knowledge base, the senior physician on the project (a superb senior clinician who had over 50 years of practice experience), other physicians, and medical students worked together, considering each of the encoded diseases. Through careful literature review and case discussions, they determined the list of pertinent findings associated with each disease. For each of these findings, they assigned a *frequency weight* (FW) and an *evoking strength* (ES), two numbers that reflect the strength of the relationship between the disease and the finding (Figure 20.5). The FW is a number between 1 and 5, where 1 means that the finding is seldom seen in the disease and 5 means it is essentially always seen (Table 20.1). The ES reflects the likelihood that a patient with the finding has the disease in question and that the disease is the cause of the finding (Table 20.2). An ES of 0 means that the disease would never be considered as a diagnosis on the basis of this finding alone, whereas an ES of 5 means that the finding is *pathognomonic* for the disease (i.e., all patients with the finding have the disease).

In addition, each finding in the knowledge base is associated with a third number, an *import number* that has a value between 1 and 5 (Table 20.3). The import number
Figure 20.5. A sample disease profile from Internist-1. The numbers beside the findings represent the evoking strength ([ES] ranging from 0 [nonspecific] to 5 [pathognomonic]) and the frequency weight ([FW] ranging from 1 [rare] to 5 [always seen]). Only an excerpt from the disease profile for echinococcal cysts is shown here.

Table 20.1. Interpretation of frequency weights.

<table>
<thead>
<tr>
<th>Frequency weight</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Listed manifestation occurs rarely in the disease</td>
</tr>
<tr>
<td>2</td>
<td>Listed manifestation occurs in a substantial minority of cases of the disease</td>
</tr>
<tr>
<td>3</td>
<td>Listed manifestation occurs in roughly one-half of the cases</td>
</tr>
<tr>
<td>4</td>
<td>Listed manifestation occurs in the substantial majority of cases</td>
</tr>
<tr>
<td>5</td>
<td>Listed manifestation occurs in essentially all cases—that is, it is a prerequisite for the diagnosis</td>
</tr>
</tbody>
</table>


Table 20.2. Interpretation of evoking strengths.

<table>
<thead>
<tr>
<th>Evoking strength</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Nonspecific—manifestation occurs too commonly to be used to construct a differential diagnosis</td>
</tr>
<tr>
<td>1</td>
<td>Diagnosis is a rare or unusual cause of listed manifestation</td>
</tr>
<tr>
<td>2</td>
<td>Diagnosis causes a substantial minority of instances of listed manifestation</td>
</tr>
<tr>
<td>3</td>
<td>Diagnosis is the most common, but not the overwhelming, cause of listed manifestation</td>
</tr>
<tr>
<td>4</td>
<td>Diagnosis is the overwhelming cause of listed manifestation</td>
</tr>
<tr>
<td>5</td>
<td>Listed manifestation is pathognomonic for the diagnosis</td>
</tr>
</tbody>
</table>

captures the idea that some abnormalities have serious implications and must be explained, whereas others may be safely ignored. Internist-1 uses the import number to handle red herrings (minor problems that are not explained by the current disease process). This familiar clinical-diagnosis problem is not handled well by formal statistical approaches.

Based on these simple measurements, Internist-1 used a scoring scheme that is similar to the hypothetico-deductive approach described in Chapter 2. The physician-user would enter an initial set of findings, and then the program would determine an initial differential diagnosis. Based on the current set of hypotheses, the program would select appropriate questions to ask, choosing from several strategies, depending on how many diseases are under consideration and how closely matched they are to the available patient data. The program considered the cost and risks of tests, as well as the benefits, and asked for simple historical and physical-examination data before recommending laboratory tests or invasive diagnostic procedures. An important feature, not previously implemented in diagnostic programs, was Internist-1’s ability to set aside some of the findings not well explained by the current differential diagnosis and to return to them later after making an initial diagnosis. Thus, Internist-1 could diagnose multiple coexistent diseases and did not make the assumptions of mutual exclusivity and completeness that have characterized most Bayesian diagnostic programs.

Using these simple knowledge structures and weighting schemes, Internist-1 demonstrated impressive diagnostic performance. In one study, the developers tested the program on 19 difficult diagnostic cases taken from a major clinical journal (Miller et al., 1982). The 19 patients had a total of 43 diagnoses, of which Internist-1 correctly identified 25. By comparison, the physicians who had cared for the patients in a major teaching hospital made 28 correct diagnoses, and the expert discussants who presented the cases before a large audience before the publication of each case in the journal correctly identified 35 diagnoses. Although Internist-1 missed several of the difficult cases (as did the physicians and discussants), the test patients had problems that were drawn broadly from across all problems in general internal medicine—no other diagnostic program would have been able to deal effectively with more than a small subset of these cases.

Internist-1 was created to run on only large, mainframe computers and therefore was not suited for widespread use by practitioners. In the 1980s, the program was adapted

---

Table 20.3. Interpretation of import values.

<table>
<thead>
<tr>
<th>Import</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Manifestation is usually unimportant, occurs commonly in normal persons, and is easily disregarded</td>
</tr>
<tr>
<td>2</td>
<td>Manifestation may be of importance but can often be ignored; context is important</td>
</tr>
<tr>
<td>3</td>
<td>Manifestation is of moderate importance but may be an unreliable indicator of any specific disease</td>
</tr>
<tr>
<td>4</td>
<td>Manifestation is of high importance and can only rarely be disregarded (as, for example, a false-positive result)</td>
</tr>
<tr>
<td>5</td>
<td>Manifestation absolutely must be explained by one of the final diagnoses</td>
</tr>
</tbody>
</table>

to run on personal computers as QMR (Quick Medical Reference) (Miller et al., 1986). Unlike Internist-1, which was developed to provide only patient-specific diagnostic advice, QMR can serve health professionals in three modes. In its basic mode, QMR is an expert consultation system that provides advice much as Internist-1 did (using essentially the same knowledge base and scoring scheme). Quick Medical Reference can also be used as an electronic textbook, listing the patient characteristics reported to occur in a given disease or, conversely, reporting which of its 600 diseases can be associated with a given characteristic. Third, as a medical spreadsheet, it can combine a few characteristics or diseases and determine the implications. For example, the user can specify two apparently unrelated medical problems and obtain suggestions about how coexisting diseases could, under the right circumstances, give rise to both problems (Figure 20.6).

The developers of QMR have argued that the system’s use as an electronic reference is far more important than its use as a consultation program to help a clinician to clinch a particularly difficult diagnosis (Miller & Masarie, 1990). In fact, in the version of QMR that was available commercially, many of the consultation features of Internist-1 were removed. For example, the QMR product did not ask questions directly of the user in order to pursue a diagnosis and did not attempt to evaluate whether more than one disease might be present at a given time.

20.5.1.2 The DXplain System

DXplain [Barnett et al., 1987; Barnett et al., 1998] is a decision support system developed at the Laboratory of Computer Science at the Massachusetts General Hospital. It was initially described by its developers as a “poor man’s Internist-1.” Despite this portrayal, the program’s capabilities are quite sophisticated. Given a set of clinical findings (signs, symptoms, laboratory data), DXplain produces a ranked list of diagnoses that might explain (or be associated with) the clinical manifestations. DXplain provides justification for why each of these diseases might be considered, suggests what further clinical information would be useful to collect for each disease, and lists what clinical manifestations, if any, would be unusual or atypical for each of the specific diseases. DXplain does not offer definitive medical consultation and, like QMR, is not intended to be used as a substitute for a human clinician. Figure 20.7 demonstrates a part of a typical DXplain session.

DXplain takes advantage of a large data base of the crude probabilities of over 4500 clinical manifestations associated with over 2000 different diseases—a knowledge base considerably larger than that of QMR. The system adopts a modified form of Bayesian reasoning to perform diagnosis using an algorithm that has not been well described in the literature. The DXplain system has been implemented both as a stand-alone version and as a server that is accessible over the Internet. The knowledge base and the user interface are continually being improved and adapted as a result of comments from users. DXplain is in use at a number of hospitals and medical schools, mostly for educational purposes, but also for clinical consultation. It is clearly the most extensively used patient-specific decision-support tool available today.

DXplain has the characteristics of both an electronic medical textbook and a medical reference system. In the role of a medical textbook, DXplain can provide
Pulmonary Disease and DIARRHEA Chronic
Pairs of diseases consistent with Entered Finding and Topic

Atelectasis
cause-by Carcinoid Syndrome Secondary to Bronchial Neoplasm

Eosinophilic Pneumonia Acute <LOEFFLER>
cause-by Hookworm Disease

Pulmonary Legionellosis
predisposed-to-by Immune Deficiency Syndrome Acquired <AIDS>

Pleural Effusion Exudative
cause-by Pancreatic Pseudocyst

Pneumococcal Pneumonia
predisposed-to-by Immune Deficiency Syndrome Acquired <AIDS>

Pulmonary Hypertension Secondary
cause-by Progressive Systemic Sclerosis
or co-occurring-with Schistosomiasis Chronic Hepatic

Pulmonary Infarction
predisposed-to-by Carcinoma of Body or Tail of Pancreas
or predisposed-to-by Carcinoma of Head of Pancreas
or cause-by Hepatic Vein Obstruction

Pulmonary Lymphoma
coinciding-with Lymphoma of Colon
or coinciding-with Small Intestinal Lymphoma

Figure 20.6. A sample associations list from QMR, a system that permits the physician to request exploratory searches of the knowledge base for associations that might be clinically relevant. For example, as shown here, the physician has asked for pulmonary diseases that may also be associated with chronic diarrhea. The resulting lists, which QMR generates dynamically, can be useful memory joggers for physicians who might otherwise overlook the suggested relationships.

Comprehensive descriptions of the more than 2000 diseases in its knowledge base, emphasizing the signs and symptoms that occur in each disease, the etiology, the pathology, and the prognosis. DXplain also provides up to 10 recent bibliographic references that have been selected as being appropriate for each specific disease. In addition, DXplain can provide a list of diseases that should be considered for any one of over
20.5.2 Patient Management: Guideline-Based Architectures and The EON System

Clinical practice guidelines are a powerful method for standardization and uniform improvement of the quality of medical care. According to the Institute of Medicine’s definition, clinical guidelines are “systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances” [Field and Lohr., 1992]. Clinical guidelines typically represent a medical expert consensus regarding the screening, diagnosis, or management, over either limited or extended periods of time, of patients who have a particular clinical problem, need, or condition (e.g., fever of unknown origin, therapy of insulin-dependent diabetes). Clinical guidelines are particularly useful for management of patients over extended periods, as in the management of chronic diseases. The application of clinical guidelines by care providers typically involves collecting and interpreting considerable amounts of data over time, applying standard therapeutic or diagnostic plans in an episodic fashion, and

Figure 20.7. A screen from a session demonstrating the DXplain diagnostic system to new users. The user had entered six findings, assisted by a standard medical vocabulary that verifies the free-text strings entered. DXplain then presents a list of the most likely common diseases and the most likely rare diseases fitting these findings. Note that DXplain also points out several pertinent findings that might or might not be present, which would further refine the diagnostic list.

5000 different clinical manifestations (signs, symptoms, and laboratory examinations) known to the system.
revising those plans when necessary. Thus, reasoning about time-oriented data and actions is essential for guideline-based care.

### 20.5.2.1 Guideline-Based Patient-Management Systems

It is now agreed that conforming to appropriate clinical practice guidelines is the best way to improve the quality of health care [Grimshaw and Russel, 1993], while reducing the escalating costs of medical care. Clinical guidelines are most useful at the point of care (typically, when the care provider has access to the patient's record), such as at the time of order entry by the care provider. In such a context, even simple reminders and alerts have powerful effects, especially in outpatient contexts. Indeed, investigators have demonstrated significant enhancement of adherence to preventive-care guidelines, such as those for the administration of pneumococcal and influenza vaccinations, by integrating several simple alerts within a hospital's order-entry system [Dexter et al., 2001].

Most clinical guidelines are text-based and not readily accessible to the healthcare workers who most need them. Even when guidelines exist in electronic format, and even when that format is available online, healthcare workers rarely have the time and means to decide which of multiple guidelines best pertains to their patients, and, if so, exactly what does applying that guideline to the particular patient entail. To support the needs of health-care providers as well as administrators, and to ensure continuous quality of care, more sophisticated information processing tools are needed. Due to limitations of current technologies, analyzing unstructured text-based guidelines is not feasible (see Chapter 8). Thus, there is an urgent need to facilitate guideline dissemination and application using machine-readable representations and automated computational methods.

There are several tasks associated with the guideline-based care that would benefit from automation. These tasks include specification (authoring) and maintenance of clinical guidelines, retrieval of guidelines appropriate to each patient, runtime application of guidelines, and retrospective assessment of the quality of the application of the guidelines. Supporting guideline-based care implies creation of a dialog between a care provider and an automated support system, each of which has its relative strengths. For example, physicians have better access to certain types of patient-specific clinical information (such as that patient's mental state and likelihood of compliance with therapy) and to general medical and commonsense knowledge. Automated systems have faster and more accurate access to detailed guideline specifications and can detect more easily prespecified complex temporal patterns in the patient's data. Thus, there is great synergy when human beings and machines work cooperatively on the problem of guideline-based care.

Developers often encode the knowledge needed to provide simple decision support for patient management, such as for raising context-sensitive alerts and reminders, as situation–action rules. An example is the rule in Figure 20.3, which is expressed in the Arden Syntax [Hripcsak et al., 1994] (Section 2.3), the earliest standard for representation and sharing of medical knowledge. A rule interpreter processes such rules—scanning the patient database for situations that trigger relevant rules, evaluating whether the condition part of the rule holds, and, if it does, taking whatever action the rule might specify. Although such rule-based approaches for representing knowledge have
been used successfully since the time of MYCIN, development and maintenance of large rule bases can be difficult. Interactions among rules may have unanticipated side effects, leading to unexpected system behaviors when rules are added to or deleted from a previously debugged knowledge base (Bachant & McDermott, 1984; Clancey, 1984; Heckerman & Horvitz, 1986). In general, rule-based approaches to the representation of clinical guidelines, such as the Arden Syntax, do not include an intuitive representation of the guideline’s clinical logic, have no semantics for the different types of clinical knowledge represented, and lack the ability to represent and reuse guidelines and guideline components. Such formalisms do not allow for inherent, intended, ambiguities in the therapy algorithm (such as when considerations exist for and against different therapy options, which need to be evaluated by the attending physician, or when the patient’s preferences need to be explicitly considered). Most important, approaches such as Arden Syntax do not support the application of guidelines over extended periods of time, as is necessary for the support of the care of patients with chronic disease. Nevertheless, rules are an excellent option when simple, one-time reminders and alerts need to be written and used, without the heavier machinery of more complex guideline representations, and in that sense they complement more expressive guideline-representation formats [Peleg et al., 2001a].

When building comprehensive decision-support systems for clinical practice guidelines, it is often most helpful to view the guidelines as reusable skeletal plans [Friedland and Iwasaki, 1985], namely a set of plans at varying levels of abstraction and detail, that, when applied to a particular patient, need to be refined by a care provider over significant time periods, while often leaving considerable room for flexibility in the achievement of particular goals. Another view is that clinical guidelines are a set of constraints regarding the process of applying the guideline (i.e., care-provider actions) and its desired outcomes (i.e., patient health states), constraints that can be viewed as the intentions of the guideline authors regarding how care should be administered and what outcomes from therapy are desirable [Shahar et al., 1998]. These constraints are often temporal, or at least have a significant temporal dimension, especially in the case of those clinical practice guidelines that concern the care of patients with chronic problems, or that specify actions to be applied over a significant period.

Numerous workers in biomedical informatics have developed several architectures for the automated application of clinical practice guidelines. Each of these systems assumes an explicit model for representing guideline knowledge. These models are similar, but make different subtle distinctions about guidelines and clinical care. It is impossible to enumerate all of the models that have been suggested, as the automation of guideline-based care remains an area of intensive investigation in biomedical informatics. Some of the best known formalisms include the ProForma language, developed by at the Advanced Computation Laboratory (ACL) of Cancer Research UK [Fox et al., 1998; Fox and Das, 2000]; the GuideLine Interchange Format (GLIF), created as part of collaboration among Columbia, Harvard, and Stanford Universities [Ohno-Machado et al., 1998; Peleg et al., 2000, 2001; Boxwala et al., 2004]; the Asbru guideline language [Miksch et al., 1997], whose focus is on highly expressive time-oriented actions, developed as part of the Asgaard project [Shahar et al., 1998]; the GUIDE model, [Quaglini et al., 2001], which is part of a more general framework, Careflow, developed at the
University of Pavia, Italy, for modeling and applying clinical guidelines in the broad context of general medical care; and the PRODIGY guideline model [Johnson et al., 2000], developed at the University of Newcastle upon Tyne in Britain. Like several of the frameworks mentioned here, the PRODIGY project uses the Protégé suite of tools to acquire and represent a set of clinical guidelines (Musen, 1998; Musen et al., 1995). Peleg and colleagues [2003] provide an excellent review of these alternative guideline-representation approaches.

20.5.2.2 A Guideline-Based Patient-Management Architecture: The EON system

Many of the current approaches for representing the knowledge of clinical-practice guidelines in computer systems build on ideas first explored in a decision-support system known as EON, which has been under development at Stanford University for nearly two decades. EON is a second-generation knowledge-based system (David et al., 1993) that aids practitioners in the care of patients who are being treated in accordance with protocols and clinical-practice guidelines (Musen, 1998; Musen et al., 1996). Unlike systems such as MYCIN or QMR, EON cannot be run by itself. Instead, EON constitutes a set of software components that must be embedded within some clinical information systems that healthcare workers use to enter and browse patient-related data. Figure 20.8 shows the major components of EON, which are the following:

![Diagram of EON architecture]

**Figure 20.8.** The EON architecture. EON consists of a number of problem-solving components (e.g., programs that plan protocol-based therapy and that determine whether a patient potentially is eligible for protocols) that share a common knowledge base of protocol descriptions. The protocol domain model, created with the Protégé system, defines the format of the protocol knowledge base. The same model also defines the schema for the database mediator, a system that channels the flow of patient data between the problem-solving components and an archival relational database. The entire architecture is embedded within a clinical information system.
Problem solvers each address specific tasks (Eriksson et al., 1995), such as (1) determining the treatment that should be given at a particular time if a patient is to receive therapy in accordance with a predefined clinical protocol and (2) determining, for a given patient, whether there are any protocols for which the patient might be eligible.

Knowledge bases encode descriptions of clinical protocols in a way that all the problem solvers in EON can examine a shared, coherent representation of protocol knowledge, thus using the protocol knowledge bases to solve their particular task. The EON therapy-planning component accesses the protocol knowledge base to identify what are the potential clinical interventions that might be administered to a given patient and what knowledge is needed to determine precisely what treatment is appropriate; the EON eligibility-determination component consults the same knowledge base to identify the factors that establish when a given protocol is appropriate and then accesses patient data to see whether the protocol might be a good match.

A database mediator serves as a conduit between all the problem solvers in EON and the database that stores all the patient data (Nguyen et al., 1997). The mediator insulates all the EON problem solvers from many of the logistical problems of querying patient data and making sense out of various time-dependent relationships among the data (e.g., querying the data for the existence of certain trends or patterns). The mediator includes within it a problem solver that addresses the specific task of abstraction of time-oriented patient data into higher-level concepts, the RÉSUMÉ system (Shahar & Musen, 1996). Thus, queries such as “Did the patient have more than 2 weeks of bone-marrow toxicity grade II (in a specific context)” can be answered by the mediator directly. The method underlying the RÉSUMÉ solver is the knowledge-based temporal-abstraction method (Shahar, 1997).

The components in EON are designed such that they can be mixed and matched to create different decision-support functionalities. For example, EON’s therapy-planning component (Tu et al., 1995) and eligibility-determination component (Tu et al., 1993), plus knowledge bases for protocol-based care of AIDS and HIV-related diseases, formed the decision-support elements of a system known as THERAPY-HELPER (or T-HELPER for short) (Musen et al., 1992). T-HELPER contained an electronic patient record with which practitioners could enter patient information at the time of each visit to an outpatient clinic specializing in the care of people with AIDS. T-HELPER would then invoke the EON components to generate specific recommendations regarding patient therapy. If the patient was not currently enrolled in applicable protocols, the T-HELPER system would indicate those protocols for which the patient potentially was eligible. For those protocols in which the patient was already enrolled, the system would indicate what therapy should be administered, given the protocol requirements, the patient’s current stage of therapy, and the patient’s clinical situation.

In other experiments, the same therapy-planning component and eligibility-determination component were used in conjunction with a knowledge base of breast-cancer protocols (Musen et al., 1996). The EON architecture made it possible to simply “plug in” these previously developed modules and to use them in conjunction with the new breast-cancer knowledge bases. Again, the EON components did not run as a stand-
alone system but were embedded within a computer-based patient-record system that could invoke the EON decision-support components when appropriate.

The EON architecture has been deployed successfully to bring guideline-based decision support to several Veterans Affairs medical centers in the United States. The EON middleware forms the basis of the ATHENA system for management of hypertension in accordance with national guidelines [Goldstein et al. 2000, 2002]. The ATHENA user interface is integrated with that of VISTA, the Veterans Affairs clinical information system, to provide an intuitive overview of the clinical data relevant to treatment of high blood pressure (Figure 20.9). The interface also displays specific suggestions to physicians regarding clinical interventions that they might make to ensure that care is consistent with the hypertension guidelines stored in the program’s knowledge base. The ATHENA knowledge base stores guideline eligibility criteria, risk stratification, blood pressure targets, relevant comorbid diseases, guideline-recommended drugs, and criteria for treatment selection and modification.

The modularity of the EON architecture makes it relatively straightforward to add new problem-solving components in addition to new knowledge bases. For example, developers might design a new problem solver that analyzed the electronic patient record retrospectively to determine whether past treatment was consistent with protocol guidelines. The new module would then be driven by the same shared knowledge bases used by the other components in EON.

Figure 20.9. An example of the ATHENA system interface. ATHENA provides decision-support for the management of hypertension using the EON architecture. In the screen capture, the provider has entered the patient’s most recent blood pressure, and is offered advice based on a relevant clinical-practice guideline. (Source: M. K. Goldstein.)
Although it may be easy to apply the EON components to new kinds of protocol knowledge bases, creating such knowledge bases in the first place can be a complex task. Fortunately, knowledge-acquisition for the EON system is greatly facilitated by the previously mentioned knowledge-base-development environment known as Protégé (Musen, 1998; Musen et al., 1995). Protégé provides a set of tools and a principled methodology for building knowledge-based systems, of which EON is only one example. Use of Protégé begins when developers create an abstract model of the application area for which knowledge-based systems are to be built. As shown in Figure 20.8, there is a common model for all the clinical-protocol knowledge bases processed by EON (Tu & Musen, 1996). This protocol model, or ontology, specifies the concepts necessary to define clinical protocols in a given domain of medicine. For example, construction of the ATHENA system required creation of a model that defined the concepts common among guidelines for hypertension (drug therapy, laboratory tests, and so on); analogously, construction of the decision-support system for breast-cancer protocols required creation of a somewhat similar model that defined the concepts common among protocols for breast cancer (including concepts such as surgery, radiotherapy, and so on). The terms and relationships of such models when entered into Protégé do more than define the concepts that form the structure of clinical protocols in machine-understandable form: The models serve as the starting point for generation of special-purpose computer-based tools that assist developers in the construction and maintenance of detailed protocol knowledge bases (Musen, 1998; Tu et al., 1995).

In the Protégé approach, developers first create a general model of the concepts and relationships that characterize a particular application area. For example, the model shown in Figure 20.10 represents a small subset of the concepts needed to define clinical protocols. A module in the Protégé system takes as input such a model and generates as output a custom-tailored tool based on that model that developers can use to enter detailed knowledge bases (Eriksson et al., 1994). Protégé thus processes the model for clinical protocols (see Figure 20.10) to construct a tool that knowledge-base authors can use for entry and review of specific protocol descriptions (Figure. 20.11). This tool—because it reflects the predefined model of clinical protocols—can be used by medical experts themselves. Because the tool is produced directly from the protocol model, developers can update and enhance the model and then generate a new knowledge-acquisition tool that reflects the corresponding changes. At the same time, developers can modify their abstract protocol models to reflect modalities of clinical care in new areas of medicine and then generate knowledge-acquisition tools that healthcare workers can use to enter new protocol specifications for these new clinical disciplines.

The EON approach demonstrates several dimensions of modern clinical decision-support systems. EON shows how decision-support systems can be embedded within larger clinical information systems. The architecture also exemplifies the use of emerging standards for network-based communication among software modules—a trend in software engineering that will become increasingly important in the years ahead (see Chapter 4). The coupling between the EON decision-support architecture and the Protégé knowledge-acquisition framework exemplifies the use of special-purpose tools for entering and maintaining protocol knowledge bases. As work in the development of clinical decision-support systems evolves, there will be expanding expectations that
decision-makers themselves be able to review and modify the electronic knowledge bases that codify an institution’s decision-making policies. As demonstrated by EON, that kind of direct involvement of clinical personnel in knowledge-base management will require increasing use of systems such as Protégé, which can simplify the creation and modification of the necessary knowledge-editing tools.

20.6 Decision Support in the Decade to Come

After more than four decades of research on medical decision-support systems, investigators have learned a great deal about the difficulties inherent in the task and about the complex barriers to successful implementation of programs. In the 1970s and 1980s, researchers made major progress in tackling the scientific questions of computer-based decision support. Technological advances in knowledge representation, knowledge
acquisition, and automated reasoning led to significant new insights related to the modeling, encoding, and dissemination of human expertise in a machine-processable form. Although early workers in biomedical informatics frequently raised concerns that clinicians might always be reluctant to interact with computer-based decision-support systems, the landscape changed radically at the end of the last century.

The advent of managed health care in the United States and growing concerns about the cost and quality of patient care globally have altered the practice of medicine in profound ways (see Chapter 23). Clinical practice guidelines based on empirical medical evidence are now ubiquitous. Practitioners often are highly motivated to follow such guidelines: Not only do physicians inherently want to offer their patients the best care possible based on available evidence in the literature, but often physicians’ remuneration and even their medical malpractice premiums depend on their ability to follow pre-defined guidelines. In an era when there is increasing emphasis on continuous quality

Figure 20.11. A screen from a Protégé-generated knowledge-acquisition tool for entry of breast cancer protocols. This tool is generated automatically from a domain model, part of which appears in Figure 20.8. The protocol depicted specifies the knowledge required to carry out a clinical trial that compares the effects of conventional adjuvant chemotherapy with those of high-dose chemotherapy followed by bone-marrow transplantation.
improvement via instruments such as clinical practice guidelines, decision-support systems such as those based on the EON architecture are assuming a central role in communicating guideline-based advice to all healthcare workers.

As healthcare organizations were undergoing radical change in the 1990s, computing technology was making an equally radical advance. The advent of the World Wide Web popularized computers in new ways. The World Wide Web demystified computers for many new users while providing connections among distributed processors that had not been imagined just a few year earlier. The Web provided a basis for inexpensive “thin clients” that could use a common browser to access a diverse collection of information resources in a uniform manner. Suddenly, it was almost trivial to bring a rich collection of programs directly to the point of care, greatly simplifying access to a variety of information sources and decision-support systems.

The educational potential of decision-support systems has long been recognized (Association of American Medical Colleges, 1986). New pressures to learn best practices coupled with the ubiquity of information technology, however, have greatly encouraged the use of computer-based decision aids in health-professional schools around the world. For the next generation of health-care workers, the use of information technology in most aspects of patient care probably will be taken for granted—much as it is in Star Trek.

20.6.1 Legal and Regulatory Questions

It may already have occurred to you that there are legal implications inherent in the development and use of such innovations. As mentioned in Chapter 10, formal legal precedents for dealing with clinical decision-support systems are lacking at present. Several observers have noted that a pivotal concern is whether the courts will view the systems under negligence law or product liability law (Miller et al., 1985). Under negligence law (which governs medical malpractice), a product or activity must meet reasonable expectations for safety. The principle of strict liability, on the other hand, states that a product must not be harmful. Because it is unrealistic to require that decision-support programs make correct assessments under all circumstances—we do not apply such standards to physicians themselves—the determination of which legal principle to apply will have important implications for the dissemination and acceptance of such tools. A related question is the potential liability borne by physicians who could have accessed such a program, and who chose not to do so, and who made an incorrect decision when the system would have suggested the correct one. As with other medical technologies, precedents suggest that physicians will be liable in such circumstances if the use of consultant programs has become the standard of care in the community (see Chapter 10). Several guidelines have been suggested for assigning legal liability to builders of knowledge-based medical decision-support systems or to the physicians using them [Allaërt & Dusserne, 1992].

Questions have also arisen regarding the validation of decision-support tools before their release (see Chapter 11). The evaluation of complex decision-support tools is challenging; it is difficult to determine acceptable levels of performance when there may be disagreement even among experts with similar training and experience. There is often no
such thing as *the* correct answer to a clinical question. Moreover, component-based architectures such as EON may comprise multiple (potentially fallible) problem-solvers, each of which addresses different clinical problems and which shares a common (potentially fallible) set of knowledge bases. The objective then becomes to isolate the cause of possible errors and to make appropriate modifications to the system overall. Evaluations of medical decision-support tools have suggested a variety of methods for assessing the adequacy of clinical knowledge bases and problem-solving components before such software is introduced for routine use (Friedman & Wyatt, 1997a).

What then should be the role of government in prerelease regulation of medical software? Current policy of the Food and Drug Administration (FDA) in the United States indicates that such tools will not be subject to federal regulation if a trained practitioner is assessing the program’s advice and making the final determination of care (Young, 1987). This policy is subject to ongoing reevaluation, however (see Chapter 10). Programs that make decisions directly controlling the patient’s treatment (e.g., closed-loop systems that administer insulin or that adjust intravenous infusion rates or respirator settings; see Chapter 17) are viewed as medical devices subject to FDA regulation.

Additional problems arise when considering suggestions to enable access to electronic patient records through the World Wide Web, such as privacy and security. Ease of access versus data security will always constitute a trade-off in clinical information systems.

### 20.6.2 Future Directions for Clinical Decision-Support Systems

Trends for decision-support research and development in the decades ahead are becoming evident. As already mentioned, the World Wide Web will continue to expand the influence of computers in all aspects of society, and the Internet and numerous intranets will link information technologies throughout large healthcare organizations and communities of patients. The Internet will bring decision-support systems designed for patient use directly into those patients’ homes and will provide more effective communication among all participants in the healthcare system (see Chapter 14). We already see many communities of patients with illnesses such as coronary-artery disease, AIDS, and breast cancer turning to the Internet to seek out the latest available information and to converse electronically both with healthcare personnel and with other patients.

Research laboratories will continue to explore how new Internet-based media can assist in clinical decision-making and how cyberspace affects both patient and provider information access and use. For example, it is unknown whether many of the information resources available on-line lead to more informed decision-making on the part of patients or whether these resources cause increased confusion and potential patient anxiety about the complexity of clinical problems.

The emerging ubiquity of the Internet in modern culture will affect the underlying technology with which decision-support systems are developed. Many research laboratories already are studying ways in which decision-support software can be assembled from previously created, tested, and debugged components—much like the software
components in the EON system. In the future, repositories of such components will be stored in libraries accessible over the Internet. Such libraries will contain, for example, reusable problem-solving modules for tasks such as diagnosis and planning that developers will apply to the construction of new decision aids. Internet-based libraries will contain standard, controlled terminologies, as well as knowledge bases of commonly needed concepts (e.g., anatomical relationships, temporal properties of clinical data, and frequently prescribed medications and their indications and side effects). Construction of clinical decision aids will involve searching the Internet-based libraries for appropriate reusable components and configuring those components into useful problem-solvers. There will be a need for a new kind of information-retrieval technology—one that can help system builders to locate and configure appropriate decision-support software components from the diverse component libraries that research laboratories, healthcare organizations, and a new industry of component vendors will make available.

Considerations of whether specific components happen to use pattern-recognition methods, Bayesian reasoning, or AI techniques will become less important, as researchers create new approaches for combining different reasoning methods to meet the specific requirements of increasingly complex decision-making tasks. Thus, by mixing and matching components, developers will be able to use Bayesian reasoners for performing probabilistic classification, AI techniques for tasks such as planning or constraint satisfaction, and mathematical models for solving problems that can be best understood in terms of systems of equations. There will be enhanced emphasis on modeling the overall task that a decision-support system performs (e.g., tasks such as therapy planning, differential diagnosis, and simulation of surgical interventions). These task models will then inform the selection of appropriate problem-solving components from various libraries.

Concomitantly, heightened understanding of organizational behavior and of clinical workflow will stimulate a new generation of clinical information systems that will integrate smoothly into the practices of healthcare workers of all kinds. Most important, these new information systems will become the vehicles for delivery of decision-support technology in the decade ahead. The very concept of a decision-support system itself will fade away, as intelligent assistants that can enhance the judgment of healthcare workers blend into the infrastructure of healthcare delivery. Automated decision support will take place with every practitioner’s routine access to clinical data in a manner that is unobtrusive, transparent, and tailored to the specific patient situation.

### 20.6.3 Conclusions

The future of clinical decision-support systems inherently depends on progress in developing useful computer programs and in reducing logistical barriers to implementation. Although ubiquitous computer-based decision aids that routinely assist physicians in most aspects of clinical practice are currently the stuff of science fiction, progress has been real and the potential remains inspiring. Early predictions about the effects such innovations will have on medical education and practice have not yet come to pass (Schwartz, 1970), but growing successes support an optimistic view of what
technology will eventually do to assist practitioners with processing of complex data and knowledge. The research challenges have been identified much more clearly, and the implications for health-science education are much better understood. The basic computer literacy of health professional students can be generally assumed, but health-science educators now must teach the conceptual foundations of biomedical informatics if their graduates are to be prepared for the technologically sophisticated world that lies ahead.

Equally important, we have learned much about what is not likely to happen. The more investigators understand the complex and changing nature of medical knowledge, the clearer it becomes that trained practitioners will always be required as elements in a cooperative relationship between physician and computer-based decision tool. There is no evidence that machine capabilities will ever equal the human mind’s ability to deal with unexpected situations, to integrate visual and auditory data that reveal subtleties of a patient’s problem, or to deal with social and ethical issues that are often key determinants of proper medical decisions. Considerations such as these will always be important to the humane practice of medicine, and practitioners will always have access to information that is meaningless to the machine. Such observations argue cogently for the discretion of healthcare workers in the proper use of decision-support tools.

Suggested Readings


This book, written by a physician who also is a sociologist, examines the difficulty of incorporating decision-support systems into clinical workflow from an organizational perspective. The book analyzes the failures of early automated decision aids and suggests new principles for decision-system design and integration.


The authors provide a comprehensive review of 100 controlled trials of clinical decision-support systems. Their analysis shows significant effects on the behavior of health-care workers in a majority (64%) of the studies. Demonstration of effects of decision-support systems on patient outcomes is more problematic, however.


This classic article provided the first influential description of how computers might be used to assist with the diagnostic process. The flurry of activity applying Bayesian methods to computer-assisted diagnosis during the 1960s was largely inspired by this provocative article.


This paper provides an overview of reusable of ontologies and problem-solving methods as components of current-generation decision support systems.

A senior clinician from Boston wrote this frequently cited article, which assessed the growing role of computers in healthcare. Thirty years later, many of the developments anticipated by Schwartz had come to pass, although the rate of change was slower than he predicted.

Staab, S. and Studer, R. eds. (2004). Handboook on Ontologies, Berlin: Springer-Verlag. This volume provides a collection of papers that describe current work on ontologies for decision support, information management, and Semantic Web applications.

Questions for Discussion

1. Researchers in medical AI have argued that there is a need for more expert knowledge in medical decision-support systems, but developers of Bayesian systems have argued that expert estimates of likelihoods are inherently flawed and that advice programs must be based on solid data. How do you account for the apparent difference between these views? Which view is valid? Explain your answer.

2. Explain the meaning of Internist-1/QMR’s frequency weights and evoking strengths. What does it mean for a finding to have a frequency weight of 4 and an evoking strength of 2? How do these parameters relate to the concepts of sensitivity, specificity, and predictive value that were introduced in Chapters 2 and 3?

3. Let us consider how deDombal and other developers of Bayesian systems have used patient-care experience to guide the collection of statistics that they need. For example, consider the database in the following table, which shows the relationship between two findings (f_1 and f_2) and a disease (D) for 10 patients.

<table>
<thead>
<tr>
<th>Patient</th>
<th>f_1</th>
<th>f_2</th>
<th>D</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

In the table, D signifies the absence of disease D. A 0 indicates the absence of a finding or disease, and a 1 indicates the presence of a finding or disease. For example, based on the above database, the probability of finding f_1 in this population is 7/10570 percent.

Refer back to Chapters 2 and 3 as necessary in answering the following questions:
   a. What are the sensitivity and specificity of each of f_1 and f_2 for the disease D?
   b. What is the prevalence of D in this 10-person population?
   c. Use the database to calculate the following probabilities:
      \* p[f_1uD]
      \* p[f_1u,D]
c. Use the database to calculate \( p[D | f_1 \text{ and } f_2] \).
d. Use the probabilities determined in b to calculate \( p[D | f_1 \text{ and } f_2] \) using a heuristic method that assumes that findings \( f_1 \) and \( f_2 \) are conditionally independent given a disease and the absence of a disease. Why is this result different from the one in c? Why has it generally been necessary to make this heuristic approximation in Bayesian programs?

4. In an evaluation study, the decision-support system ONCOCIN provided advice concerning cancer therapy that was approved by experts in only 79 percent of cases (Hickam et al., 1985b). Do you believe that this performance is adequate for a computational tool that is designed to help physicians to make decisions regarding patient care? What safeguards, if any, would you suggest to ensure the proper use of such a system? Would you be willing to visit a particular physician if you knew in advance that she made decisions regarding treatment that were approved by expert colleagues less than 80 percent of the time? If you would not, what level of performance would you consider adequate? Justify your answers.

5. A large international organization once proposed to establish an independent laboratory—much like Underwriters Laboratory in the United States—that would test medical decision-support systems from all vendors and research laboratories, certifying the effectiveness and accuracy of those systems before they might be put into clinical use. What are the possible dimensions along which such a laboratory might evaluate decision-support systems? What kinds of problems might such a laboratory encounter in attempting to institute such a certification process? In the absence of such a credentialling system for decision-support systems, how can health-care workers feel confident in using a clinical decision aid?