22 Clinical Decision-Support Systems

Mark A. Musen, Blackford Middleton, and Robert A. Greenes

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

- What are the key motivations for clinical decision support?
- How is clinical decision support relevant to the concept of "meaningful use" of EHRs in the United States?
- What are typical design considerations when building a decision-support system?
- What are some ways in which developers of decision-support systems encode clinical knowledge?
- What are some current standards in the HIT industry that facilitate the construction of decision-support applications?
- Why has adoption been slow and what are prospects for broader use?
- What are the key areas for research and development in clinical decision-support systems?

In this chapter, we discuss information technology that assists with **clinical decision support** (CDS) – the process that "provides clinicians, staff, patients, or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care" (Osheroff et al. 2007). Systems that provide CDS do not simply assist with the retrieval of relevant information; they communicate information that takes into consideration the particular clinical context, offering situation-specific information and recommendations. At the same time, such systems do not themselves perform clinical decision making; they provide relevant knowledge and analyses that enable the ultimate decision makers—clinicians, patients, and health care organizations—to develop more informed judgments. Ideally, CDS systems may be described in terms of five *right* things that they do: they "provide the right information, to the right person, in the right format, through the right channel, at the right point in workflow to improve health and health care decisions and outcomes" (Osheroff et al. 2004).

Systems that provide CDS come in three basic varieties: (1) They may use information about the current clinical context to retrieve highly relevant

M.A. Musen, MD, PhD (✉)
Center for Biomedical Informatics Research,
Stanford University School of Medicine, 1265 Welch
Road, Room X-215, Stanford 94305-5479, CA, USA
e-mail: musen@stanford.edu

B. Middleton, MD, MPH, MSc
Informatics Center, Vanderbilt University Medical
Center, 3401 West End Ave., Suite 700,
Nashville 37203, TN, USA
e-mail: blackford.middleton@vanderbilt.edu

R.A. Greenes, MD, PhD
Department of Biomedical Informatics,
Arizona State University, Tempe, AZ, USA

Division of Health Sciences Research,
College of Medicine, Mayo Clinic, Samuel Johnson
Research Bldg, 13212 East Shea Boulevard,
Scottsdale 85259, AZ, USA
e-mail: greenes@asu.edu

We are grateful to the co-authors of the chapter on this subject that appeared in the previous edition of this text book, Drs. Yuval Shahar and Edward H. Shortliffe.

E.H. Shortliffe, J.J. Cimino (eds.), *Biomedical Informatics*,
DOI 10.1007/978-1-4471-4474-8_22, © Springer-Verlag London 2014

online documents, as with so-called "**infobuttons**" (introduced in Chap. 12); (2) they may provide patient-specific, situation-specific alerts, reminders, physician order sets, or other recommendations for direct action; or (3) they may organize and present information in a way that facilitates problem solving and decision making, as in **dashboards**, graphical displays, documentation templates, structured reports, and order sets. Order sets are a good example of the latter because they both may facilitate decision making by providing a mnemonic function and also may enhance workflow by providing a means to select a group of relevant activities quickly. As we discussed in Chap. 1, many observers consider knowledge resources that distill the medical literature and that facilitate manual selection of content relevant to the current situation to be simple decision-support systems.

This chapter provides a motivation for computer-based decision aids, emphasizing the current health care situation in the United States while keeping an eye on global trends. It offers some historical background regarding CDS systems, then provides a description of current implementation strategies and challenges, and closes with discussion of critical research questions that must be addressed to ensure optimal effectiveness of CDS in clinical practice.

22.1 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, people have recognized, from the earliest days of computing, that computers might support health-care workers by helping these people to sift through the vast collections of possible diseases, findings, and treatments.

We can view the contents of this entire book as addressing clinical data and decision-making. In Chap. 2, we discussed the central role of accurate, complete, and relevant data in supporting the decisions that confront clinicians and other health-care workers. In Chap. 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 Chap. 4 we introduced the cognitive issues that underlie 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 indirect effect on the quality of health care decisions. In this chapter, we bring together these themes by concentrating on methods and systems that have been developed specifically to assist health workers in making 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 Chaps. 3 and 4, is the **diagnostic process**—deciding which questions to ask, tests to order, or procedures to perform, and assessing the value of the results that can be obtained in relation 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? (In that sense, the response to treatment is also a type of diagnostic test). Lastly, when a clinician and a patient are faced with alternative treatments, and they seek help to choose among them, the estimation of prognosis for cure or risk of death or complications is an important decision-making activity.

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 Chap. 27, for example, we examine the need for formal decision techniques and tools in creating health policies. As we develop databases that can identify patients with specific diseases, with risks of complications, or in need of specific interventions such as screening tests or immunizations (see Chap. 16), so-called **population management** can be used to provide a form of decision support for groups of patients. Some clinical decision support is also aimed directly at patients, in terms of alerts, reminders, or aids to interpretation of information; techniques for assessing prognosis and risk of alternative strategies should involve shared decision making between providers and patients, which is also an important area of activity.

In this chapter, we focus on decision aids for the provider in particular. 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 patient must be adequate for making an informed decision, but they must not be excessive (see Chap. 4). 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, Chap. 19). Thus, it is important to know when additional data will confuse rather than clarify and when it is imperative to use tools (computational, visual, or otherwise) that permit data to be summarized for easier cognitive management (see Chap. 4). Operating rooms and intensive-care units are classic settings in which this problem arises; 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 Chap. 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 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 those where a more prescriptive 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 tradeoffs 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. Similar issues relating to test or treatment selection, in terms of costs, risks, and benefits, must be understood. Awareness of biases and of the ways that they can creep into problem-solving also are crucial (see Chap. 3). 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 CDS. The program must have access to good data, it must have extensive background knowledge encoded for the clinical domain in question, and it must embody an intelligent approach to problem-solving that is sensitive to requirements for proper analysis, appropriate cost–benefit tradeoffs, and efficiency.

22.2 Motivation for Computer-Based CDS

Since the 1960s, workers in biomedical informatics have been interested in CDS systems both because of a desire to improve health care and to understand better the process of medical decision-making. Building a computer system that attempts to process data as a clinician does provides insight into the nature of medical problem solving and enables the creation of formal models of clinical reasoning. At the same time, construction of such systems offers obvious societal benefits if the computer programs can aid practitioners in their care of patients and lead to better clinical outcomes. Although the more academic considerations have provided strong motivation for work in the area of computer-based decision aids over several decades, the recognition of the importance of **clinical decision support systems** (**CDSSs**) as practical tools has increased markedly in recent years as a result of the inexorable growth in health care complexity and cost, as well as the introduction of health care legislation aimed at addressing these trends—which have made the development and broad adoption of CDS technology a priority.

The twenty-first century has seen changes in health care practices that make the development of CDS technology particularly necessary. Computer-based CDS has taken on increasing urgency for three reasons: (1) increasing challenges related to knowledge and information management in clinical practice, (2) the pressure to adopt and meaningfully use electronic medical records, and (3) the goal of delivering increasing personalized health care services – tailored to the patient’s preferences for care and to his or her individual genome. We consider these three factors in turn.

22.2.1 Physician Information Needs and Clinical Data Management

Modern clinical practice is characterized by an ever-expanding knowledgebase in clinical medicine, and by a growing clinical data set describing every patient characteristic from phenotype to genotype (Kohn et al. 2002). Despite the growing amounts of data and knowledge with which physicians need to work, health care workers have seen the average time for a clinical encounter steadily decrease, particularly in the United States, where the pressures of the prevalent fee-for-service reimbursement system and a concomitant rise in the amount of paperwork required for administrative management and billing continue to squeeze practitioners (Baron 2010). Studies of information needs among physicians in clinical practice have long revealed that unanswered clinical questions are common in ambulatory clinical encounters, with as many as one or two unanswered clinical questions about diagnosis, therapy, or administrative issues arising in every visit (Covell et al. 1985). In as many as 81 % of clinical encounters in ambulatory care, clinicians may be missing critical information, with an average of four missing items per case (Tang et al. 1994b, 1996). Providers consequently face major challenges in accessing relevant information, acquiring a complete picture of the patient’s clinical state and history, and knowing what further testing or therapeutic actions are best to take. Studies suggest that as many as 18 % of medical errors may be due to inadequate availability of patient information (Leape 1994). The demands for increased information management in the setting of an ever expanding clinical knowledge base are primary drivers for the adoption of CDS systems. (See Chap. 21 for a deeper discussion of physician information needs.)

22.2.2 EHR Adoption and Meaningful Use

These challenges, coupled with the seemingly inexorable rise in health care costs, have led to a variety of cost-containment and quality-improvement strategies in recent years. Health care delivery in the United States is in the midst of a profound transformation, in part due to Federal public policy efforts to encourage the adoption and use of health information technology (HIT). The American Recovery and Reinvestment Act (ARRA) of 2009, and the **HITECH regulations** within it, created incentives for the widespread adoption of health information technologies (Blumenthal 2009; see Chap. 27). These public policy efforts are often viewed as an essential adjunct to current health payment reform efforts in the United States, and a prelude to additional health care delivery redesign, payment reform, and cost containment. Even as recently as 2012, only 34.8 % of physicians in ambulatory practice in the United States used a basic or comprehensive electronic medical record (Decker et al. 2012), and 26.6 % of U.S. hospitals used health information technologies in inpatient care-delivery settings (DesRoches et al. 2012), although these numbers are on a rapid upward trajectory. The ARRA and HITECH policies, and the resulting technology adoption, are changing the practice of medicine and clinical care delivery in both beneficial and untoward ways (Sittig and Singh 2011). To achieve meaningful and effective use of HIT, the software must be viewed as one component of a complex sociotechnical system, in which all elements must work effectively (Institute of Medicine 2011a).

One of the principal motivations for EHR adoption is to provide an infrastructure with which to improve the quality, safety, and efficacy of health care delivery. In recent years, the U.S. government has placed considerable emphasis on the adoption of quality measures and quality-reporting requirements as part of meaningful use of HIT (Clancy et al. 2009; Institute of Medicine 2011a). Quality measures, despite their ability to provide feedback that stimulates improved performance by the clinician, are only part of the process needed to make the desired improvements. Prospective, proactive clinical decision support must also be in place. The U.S. government's rules for **meaningful use** of HIT have required only minimal CDS compliance at the time of this writing, but Phase III of the meaningful use regulations in 2016 is expected to increase the mandate for CDS in EHR systems substantially (Blumenthal and Tavenner 2010; see Chap. 27).

22.2.3 Personalized Medicine

The fundamental model for the practice of medicine has undergone dramatic change in the past century or so. The objectives of clinical care have shifted radically from the archaic goal of correcting putative imbalances of bodily humors to the scientific understanding of pathophysiology and of mechanisms for eliminating pathogens and for remedying biological aberrancies. The resulting view of medicine as the application of biological principles was at the core of the report produced by Abraham Flexner (1910) that upended medical education in the early twentieth century and that had led to the **reductionist biomedical model** that prevailed for the rest of that century. More recently, however, George Engel's **biopsychosocial model** (Engel 1977) brought to the fore of clinical care the need to address psychological and social factors in clinical treatment plans in addition to underlying biomedical problems. By the end of the twentieth century, it became increasingly accepted that CDS requires not only communication of scientific medical knowledge, but also adaptation of that knowledge to reflect the psychological and social situation that would temper the application of the knowledge. Added to this complexity is the aging of the population, owing in part to advances in health and health care, resulting in a much higher burden of chronic diseases, multiple diseases, and multiple testing and treatment options, with both their positive and negative consequences that must be balanced—all contributing to the increasing intricacy of care.

As a further extension of these trends, the genomic era in which we now live has further increased the need for clinical practice to reflect **personalized medicine** and the need to tailor care to individual factors in ways that never before were imaginable (Ginsburg and Willard 2009). Personalized medicine is characterized by decision making that takes into account and that is specific to patient personal history, family history, social and environmental factors, along with genomic data and patient preferences regarding their own care (see Chap. 25). In this approach, clinical decision making is explicitly patient-centered in new ways, bringing the best evidence at the genetic level to bear on many clinical scenarios, while incorporating patient preferences for acquiring and applying genetic information (Fargher et al. 2007). Personalized genetic medicine (Chan and Ginsburg 2011) is already generating data that outstrip the information and knowledge processing capabilities of practitioners, and many clinicians feel threatened by the impending tsunami of additional knowledge that they will need to master (Baars et al. 2005). As personalized medicine becomes the norm, primary-care and specialist practitioners alike will need to manage their patients by interpreting genomic tests along with myriad other data at the point of care. It is hard to imagine how clinicians will manage to perform such activities without substantial computer-based assistance. Informatics is well suited to support a personalized approach to clinical genetic medicine (Ullman-Cullere and Mathew 2011).

Another related change is growing recognition of the importance of promoting optimal health and wellness, not just by treating disease but by encouraging healthy lifestyles, fostering compliance with health and health care regimens, and carrying out periodic health-risk assessments. Key to personalized medicine will be tools to support such *prospective medicine* (Langheier and Snyderman 2004)—assisting the acquisition of a detailed family history, social history, and environmental history, providing health-risk assessments, and managing genomic information (Hoffman and Williams 2011; Overby et al. 2010).

22.2.4 Savings Potential with Health IT and CDS

CDS has been shown to influence physician behavior (Colombet et al. 2004; Lindgren 2008; Schedlbauer et al. 2009), diagnostic test ordering and other care processes (Bates and Gawande 2003; Blumenthal and Glaser 2007), and the costs of care (Haynes et al. (2010)), and it may have a modest impact on clinical outcomes (Bright et al. 2012). While there is great promise with HIT and CDS, their implementation is not without potential peril: HIT poorly designed or implemented, or misused, can generate unintended consequences (Ash et al. 2007; Harrison et al. 2007; Bloomrosen et al. 2011), and introduce new types of medical errors (Institute of Medicine 2011a).

Only a handful of studies have examined the **return on investment** (ROI) for HIT, and even fewer have investigated that for decision-support specifically. The value of CDS in terms of ROI is difficult to measure. Isolated studies of various hand-crafted systems in academic centers have shown value, but adoption elsewhere has often been problematic. Broad adoption has not occurred, for many reasons discussed later in this chapter, including the proprietary nature of systems for CDS and for representation of knowledge, the lack of interoperability of data, the mismatch of CDS to workflow, and usability concerns.

Systematic reviews of the scientific literature, such as the one performed by Bright and colleagues (Bright et al. 2012), have not been able to demonstrate an effect of CDS on patient outcomes except in the short term. This finding is not surprising, however, because the time point at which CDS occurs is often long before a final outcome, and many intervening factors may have a greater effect. In the case of CDS for many chronic diseases whose complications ensue over years or decades, it simply may be impractical to continue longitudinal studies long enough to be able to measure meaningful differences in outcome.

Historically, the adoption of CDS technology has been motivated by a virtuous desire to enhance the performance of clinicians when dealing with complex situations. The recent advent of legal, regulatory, and financial drivers, as well as the increasing importance of personalizing medical decision making on the basis of genomic data, now make CDS an essential element of modern clinical practice.

22.3 Methods of CDS

As we have already noted, CDS systems (1) may use information about the current clinical context to retrieve pertinent online documents; or they (2) may provide patient-specific, situation-specific alerts, reminders, physician order sets, or other recommendations for direct action; or they (3) may organize information in ways that facilitate decision making and action. Category (2) largely consists of the various computer–based approaches ("classic" CDS systems) that have been the substrate for work in biomedical informatics since the advent of applied work in probabilistic reasoning and artificial intelligence in the 1960s and 1970s. Such systems provide custom-tailored assessments or advice based on sets of patient-specific data. They may follow simple logics (such as algorithms), they may be based on decision theory and cost–benefit analysis, or they may use probabilistic approaches only as an adjunct to symbolic problem solving. Some diagnostic assistants (such as DXplain Barnett et al. 1987) suggest differential diagnoses or indicate additional information that would help to narrow the range of etiologic possibilities. Other systems suggest a single best explanation for a patient's symptomatology. Other systems interpret and summarize the patient's record over time in a manner sensitive to the clinical context (Shahar and Musen 1996). Still other systems provide therapy advice rather than diagnostic assistance (Musen et al. 1996).

It is helpful to review some of the early work on such systems to get a sense of the scientific questions that need to be addressed in order to build CDS systems and to understand the challenges that currently confront the field.

Since the earliest days of computers, health professionals have anticipated the time when machines would assist them in the diagnostic process. The first article dealing with this possibility appeared in the late 1950s (Ledley and 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 lack of availability of needed data and 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 CDS systems and demonstrate paradigms for CDS implementation that still are prevalent today. These decision aids are de Dombal's system for diagnosis of abdominal pain (de Dombal et al. 1972), Shortliffe's MYCIN system for selection of antibiotic therapy (Shortliffe 1976), and the HELP system for delivery of inpatient medical alerts (Kuperman et al. 1991; Warner 1979). We emphasize these three artifacts not because any of them has had a durable effect on clinical practice, but because they each demonstrate very well defined principles for automated decision making that, in their own ways, continue to inspire modern CDS systems that are more complex and more eclectic in their computational architectures.

22.3.1 Leeds Abdominal Pain System

Starting in the late 1960s, F. T. de Dombal and his associates at the University of Leeds studied the diagnostic process and developed computer-based decision aids using Bayesian probability theory (see Chap. 3). Using surgical or pathologic diagnoses as the gold standard, they emphasized the importance of deriving the conditional probabilities used in Bayesian reasoning from high-quality data that they gathered by collecting information on thousands of patients (Adams et al. 1986). Their system, the Leeds abdominal pain system, used sensitivity, specificity, and disease-prevalence data for various signs, symptoms, and test results to calculate, using Bayes' theorem, the probability of seven possible explanations for acute abdominal pain (appendicitis, diverticulitis, perforated ulcer, cholecystitis, small-bowel obstruction, pancreatitis, and non-specific abdominal pain). To keep the Bayesian computations manageable, the program made the assumptions of (1) conditional independence of the findings for the various diagnoses and of (2) mutual exclusivity and exhaustiveness of the seven diagnoses (see Chap. 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 emergency room with abdominal pain of sudden onset. The data from these sheets provided 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 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 (1) to the diagnoses reached by the attending clinicians and (2) to the ultimate diagnosis verified during surgery or through appropriate tests (the "gold standard"; see Chap. 2).

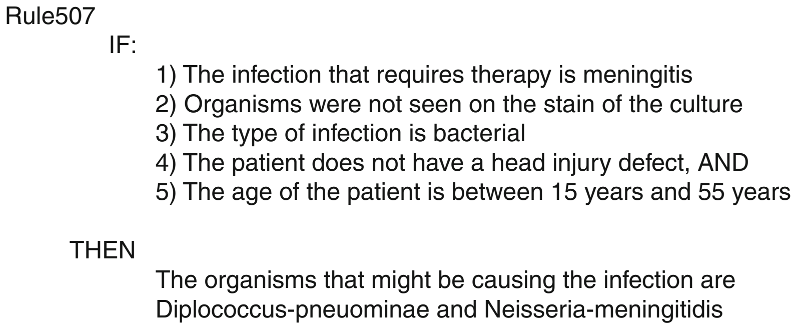
In contrast to the clinicians' diagnoses, which were correct in only 65–80 % of the 304 cases (with accuracy depending on the individual clinician's training and experience), the program's diagnoses were correct in 91.8 % of cases. Furthermore, in six of the seven disease categories, the computer was 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 6 patients who did have appendicitis were observed for more than 8 h before they were finally taken to the operating room.

With the introduction of personal computers, de Dombal'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."[1] Another possible explanation is that there are different probabilistic relationships between findings and diagnoses in different patient populations.

22.3.2 MYCIN

A different approach to computer-assisted decision support was embodied in the MYCIN program, a rule-based consultation system that combined diagnosis with 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 underlying knowledge 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 use of interacting rules to represent knowledge about organisms that might be causing a patient's infection and the antibiotics that might be used to treat it.

[1] *Rebound tenderness* is pain that is exacerbated when the physician presses down on the abdomen and then suddenly releases, generating a "rebound" when the abdomen returns to its baseline position.



Rule507

IF:

1) The infection that requires therapy is meningitis
2) Organisms were not seen on the stain of the culture
3) The type of infection is bacterial
4) The patient does not have a head injury defect, AND
5) The age of the patient is between 15 years and 55 years

THEN

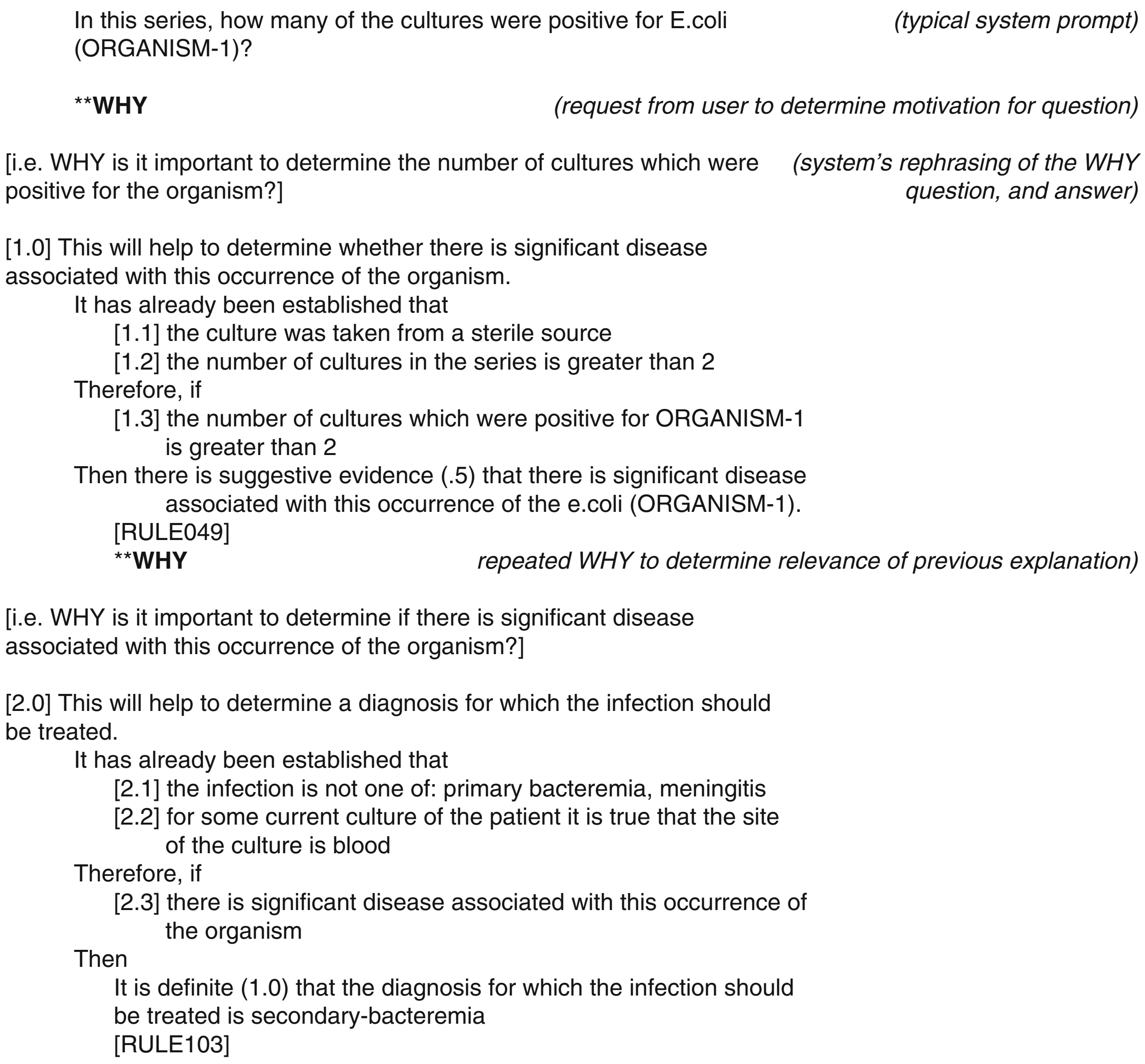
The organisms that might be causing the infection are
Diplococcus-pneuominae and Neisseria-meningitidis

Fig. 22.1 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

Knowledge of infectious diseases in MYCIN was represented as production rules (Fig. 22.1). A production rule is simply a conditional statement that relates observations to associated inferences that can be drawn. The conclusions drawn by one production rule may be considered as input observations by other rules when a system of rules is used for reasoning. 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 MYCIN reasoning program used an approach called **backward chaining**; whenever a rule was being considered such that the system did not know whether the condition on the left-hand side of the rule (i.e., the premise) was true, MYCIN would look backward to see whether the knowledge base contained any other rules that, when evaluated, could conclude information that might inform the evaluation of the current rule's premise. (Nearly all contemporary rule-based systems, on the other hand, use an inference method known as **forward chaining**: Whenever a production rule "fires" and the conclusion of that rule is proven to be true, the system looks forward for other rules in the rule base that could also might be able to fire now that the new conclusion is known to be true.)
- The rules often formed a coherent explanation of MYCIN's reasoning—those that applied to the current decision were displayed in response to a user's questions (Fig. 22.2). 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 significant, unintended side effects.

The developers evaluated MYCIN's performance on therapy selection for patients with blood-borne bacterial infections (Yu et al. 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 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. Work on MYCIN helped to clarify how techniques known as **knowledge acquisition** could help developers to work with subject-matter experts to build new rule bases, how general-purpose rule interpreters could help offer decision support for new application areas when developers swapped in a new rule base, and why computer systems tailored to offer CDS are not always capable of offering their users compelling explanations of their decision logic.



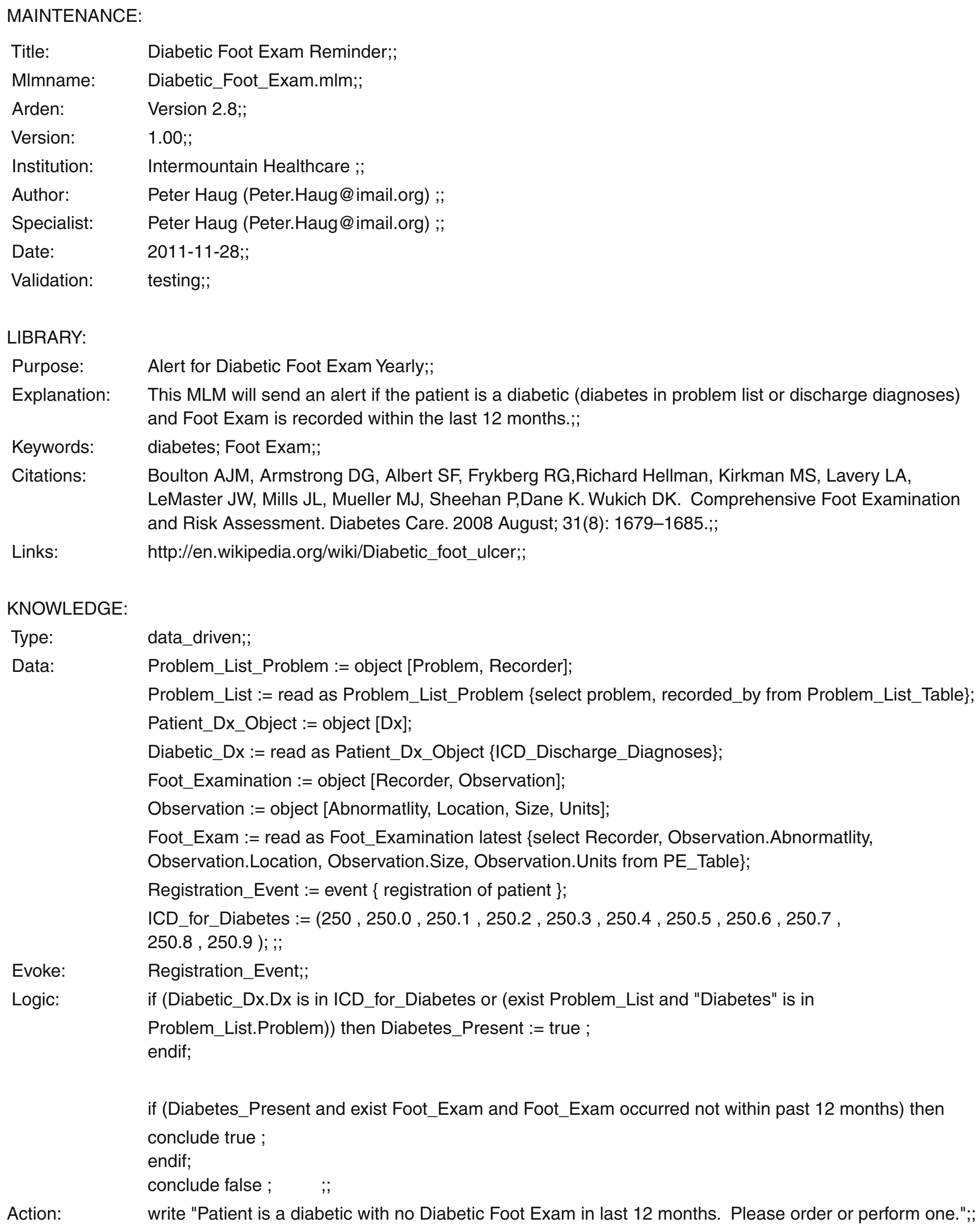
```
          In this series, how many of the cultures were positive for E.coli          (typical system prompt)
          (ORGANISM-1)?

          **WHY                         (request from user to determine motivation for question)

     [i.e. WHY is it important to determine the number of cultures which were   (system's rephrasing of the WHY
     positive for the organism?]                                                          question, and answer)

     [1.0] This will help to determine whether there is significant disease
     associated with this occurrence of the organism.
          It has already been established that
               [1.1] the culture was taken from a sterile source
               [1.2] the number of cultures in the series is greater than 2
          Therefore, if
               [1.3] the number of cultures which were positive for ORGANISM-1
                    is greater than 2
          Then there is suggestive evidence (.5) that there is significant disease
                    associated with this occurrence of the e.coli (ORGANISM-1).
               [RULE049]
               **WHY                  repeated WHY to determine relevance of previous explanation)

     [i.e. WHY is it important to determine if there is significant disease
     associated with this occurrence of the organism?]

     [2.0] This will help to determine a diagnosis for which the infection should
     be treated.
          It has already been established that
               [2.1] the infection is not one of: primary bacteremia, meningitis
               [2.2] for some current culture of the patient it is true that the site
                    of the culture is blood
          Therefore, if
               [2.3] there is significant disease associated with this occurrence of
                    the organism
          Then
               It is definite (1.0) that the diagnosis for which the infection should
               be treated is secondary-bacteremia
               [RULE103]
```

Fig. 22.2 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

22.3.3 HELP

The HELP system, an integrated hospital information system at LDS Hospital in Salt Lake City that was built up locally during the 1970s and 1980s, provided one of the first demonstrations of the importance of integrating CDS capabilities with underlying information technology. HELP had the ability to generate automated alerts when abnormalities in the patient record were noted, and its impact on the development of the field was immense, with applications and methodologies that span nearly the full range of activities in biomedical informatics (Kuperman et al. 1991).

HELP added to a conventional medical-record system a monitoring program and a mechanism for storing decision logic in **HELP sectors** that could be viewed as rules that relate the values of data in the patient database to actions that health care workers might be reminded to take. HELP thus provided a mechanism for event-driven generation of specialized warnings, alerts, and reports. 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). In the Arden Syntax, each decision rule, or HELP sector, is called a **medical logic**



```
MAINTENANCE:
 Title:           Diabetic Foot Exam Reminder;;
 Mlmname:         Diabetic_Foot_Exam.mlm;;
 Arden:           Version 2.8;;
 Version:         1.00;;
 Institution:     Intermountain Healthcare ;;
 Author:          Peter Haug (Peter.Haug@imail.org) ;;
 Specialist:      Peter Haug (Peter.Haug@imail.org) ;;
 Date:            2011-11-28;;
 Validation:      testing;;

LIBRARY:
 Purpose:         Alert for Diabetic Foot Exam Yearly;;
 Explanation:     This MLM will send an alert if the patient is a diabetic (diabetes in problem list or discharge diagnoses)
                  and Foot Exam is recorded within the last 12 months.;;
 Keywords:        diabetes; Foot Exam;;
 Citations:       Boulton AJM, Armstrong DG, Albert SF, Frykberg RG,Richard Hellman, Kirkman MS, Lavery LA,
                  LeMaster JW, Mills JL, Mueller MJ, Sheehan P,Dane K. Wukich DK.  Comprehensive Foot Examination
                  and Risk Assessment. Diabetes Care. 2008 August; 31(8): 1679–1685.;;
 Links:           http://en.wikipedia.org/wiki/Diabetic_foot_ulcer;;

KNOWLEDGE:
 Type:            data_driven;;
 Data:            Problem_List_Problem := object [Problem, Recorder];
                  Problem_List := read as Problem_List_Problem {select problem, recorded_by from Problem_List_Table};
                  Patient_Dx_Object := object [Dx];
                  Diabetic_Dx := read as Patient_Dx_Object {ICD_Discharge_Diagnoses};
                  Foot_Examination := object [Recorder, Observation];
                  Observation := object [Abnormatlity, Location, Size, Units];
                  Foot_Exam := read as Foot_Examination latest {select Recorder, Observation.Abnormatlity,
                  Observation.Location, Observation.Size, Observation.Units from PE_Table};
                  Registration_Event := event { registration of patient };
                  ICD_for_Diabetes := (250 , 250.0 , 250.1 , 250.2 , 250.3 , 250.4 , 250.5 , 250.6 , 250.7 ,
                  250.8 , 250.9 ); ;;
 Evoke:           Registration_Event;;
 Logic:           if (Diabetic_Dx.Dx is in ICD_for_Diabetes or (exist Problem_List and "Diabetes" is in
                  Problem_List.Problem)) then Diabetes_Present := true ;
                  endif;

                  if (Diabetes_Present and exist Foot_Exam and Foot_Exam occurred not within past 12 months) then
                  conclude true ;
                  endif;
                  conclude false ;         ;;
Action:           write "Patient is a diabetic with no Diabetic Foot Exam in last 12 months.  Please order or perform one.";;
```

Fig. 22.3 This medical logic module (MLM), written in the Arden syntax, prints a warning for health care workers whenever a patient who has diabetes is registered for a clinic visit and has not had a documented foot examination in the past year. 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. The *data* slot defines the variables that are to be used by the MLM; the text between curly braces must be translated into queries on the local patient database when the MLM is deployed locally (Source: P. J. Haug, Intermountain Healthcare)

module (MLM). Figure 22.3 shows one such MLM and its representation in the Arden syntax. An MLM is a specialized form of what is known as an **Event-Condition-Action** (**ECA**) **rule**, in that evaluation of situation-specific conditional expression logic is triggered by an external event, and, if the condition evaluates to be "true", then an action is performed.

Whenever new data about a patient became available, regardless of the source, the HELP system checked to see whether the data matched the criteria for invoking an MLM. If they did, the system would evaluate the MLM to see whether that MLM was relevant for the specific patient. The logic in these MLMs was developed by clinical experts who collaborated with workers in informatics. The output generated by successful MLMs included, for example, alerts regarding untoward drug actions, interpretations of laboratory tests, or calculations of the likelihood of diseases. This output was 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 was intended.

Another important extension of the idea of alerts is *clinical reminders*, in which the triggering event is usually time – such as the age of the patient, or an elapsed time since a previous event – coupled with other conditions, first popularized and implemented widely at Regenstrief Institute of Medicine in Indianapolis, Indiana (McDonald 1976). Like the HELP system implemented at LDS Hospital, the Regenstrief Medical Information System used MLMs encoded as rules to generate one-step decision logic (McDonald 1981).

22.3.4 Comparing the Early CDS Systems

The Leeds system, MYCIN, and HELP demonstrate the most fundamental issues that developers of computer-based decision aids face: (1) identifying the input data that will drive decision making, (2) determining how the output of the CDS system will be communicated, and (3) constructing a mechanism to reason about the inputs to generate appropriate output. We will address each of these issues in detail in the remainder of this chapter. First, however, it is useful to review how these elements were addressed in each of these classic systems.

The three historical systems differed radically in how the input data were collected. In the Leeds abdominal pain system, the input was derived from a simple checklist completed by the clinician that enumerated some of the findings that a patient with abdominal pain might have. Data collection was not burdensome, since the checklist was rather short, although the results did need to be transcribed into the computer. Some observers have suggested that the availability of the checklist itself could have been responsible for many of the benefits of the Leeds system, since it reminded the clinicians of key questions that they needed to ask their patients in the first place (see Gawande 2009). On the other hand, use of the MYCIN system required a potentially lengthy question-and-answer dialog with the computer that would have to take place outside of the usual clinical workflow. The barrier imposed by this style of interaction remains a major impediment to the adoption of MYCIN-style computer-based consultation systems (although with modern EHRs, some of the data could be obtained in such a system directly from the stored record instead of being entered manually by a user). With the HELP system, of course, there are no problems of human–computer interaction; the data are already available, provided by the health information system as a function of routine care. Although the ability to drive a CDS system based on data that require no manual entry has compelling advantages, HELP had the obvious disadvantage that any data that were not in the database—or that were in the database but not available in coded or numerical form—could not be brought to bear on the decision process. Nevertheless, the integration of CDS with EHR functionality in the HELP system was an important move away from the idea of standalone "consultation systems," such as MYCIN, which might provide comprehensive and complete patient advice for a particular problem, to more opportunistic CDS technology that could use readily available, but sometimes incomplete patient data to offer recommendations in a manner that did not require clinicians to step outside their usual workflow (Miller and Masarie 1990).

The output of the Leeds system was simply a posterior probability for each of the seven diagnoses for abdominal pain for which it had been programmed. MYCIN provided the user with an elaborate description of the infections that might be present in the patient, the antibiotics that should be administered, and the possible side effects of those antibiotics. More important, MYCIN supported a **mixed-initiative dialog** through which the user could investigate the chain of inference rules that supported some aspects of the program's output. With HELP, the output was a predefined text message (sometimes customized for the patient's particular clinical situation) that would appear on a line printer at the patient's nursing station or at some other appropriate location in the hospital. In subsequent versions of HELP, as with more contemporary systems that offer alerts or reminders, such messages appeared in the form of popup windows in the EHR or emails or text messages sent directly to the provider.

The three decision aids differ significantly in the manner in which they reached their conclusions. The Leeds system used a large database of case histories to calculate the conditional probabilities of a fixed number of diseases given a fixed number of possible patient findings, and applied Bayes' theorem to specific sets of input data. MYCIN, on the other hand, eschewed the use of formal probability theory and pioneered the use of chaining production rules that interacted at runtime to deduce the possible pathogens causing infection and to suggest a treatment regimen that could provide coverage for each of these germs. The developers of MYCIN did explore a heuristic method for dealing with uncertainty, called "certainty factors," that propagated uncertainty about the conclusions of rules when the rules were chained, and which did have a relationship to subjective probability estimations (Shortliffe and Buchanan 1975). HELP adopted a rule-based approach, but its rules typically did not "chain," and rule firing was totally deterministic if the condition part of the MLM evaluated to "true" based on the pattern of findings in the patient database.

22.4 Principles of CDS System Design

Modern CDS systems typically achieve their results using Bayesian reasoning, production rules, MLMs, knowledge-based groupings of physician orders, referred to as "order sets," and other templates, or by the use of prediction associations derived by mining and analysis of EHR data (or some combination of these approaches). Like the historical programs that we reviewed in Sect. 22.3, contemporary systems may acquire the data on which they base their recommendations interactively from users or directly from a health information system (or some combination of these approaches). We now discuss the issues that drive CDS system design, and we highlight how these issues are manifest in current clinical decision aids.

22.4.1 Acquisition and Validation of Patient Data

A prerequisite to any decision making process is having available all the data that are required to perform the required actions. As emphasized in Chap. 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 acquisition, 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).

The problems of data acquisition go beyond entry or extraction from the EHR 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. As discussed in detail in Chap. 7, there are several controlled medical terminologies that health care 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 LOINC codes), and so on. Still, there is no controlled terminology that can capture all 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. The prose of progress notes, consultation notes, operation reports, discharge summaries, and other documents contains an enormous amount of information that never makes it to the coded part of the EHR. 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).

The desire to access information from the EHR that may be available only in free text has been a topic of great concern to the CDS community. Some information systems provide options for **structured data entry**, asking clinicians to use fill-in-the-blanks forms or templates on the computer screen to enter information that otherwise would be entered as part of a textual note. In general, providers have resisted such human–computer interfaces, often finding it restrictive and cumbersome to make selections from predefined menus when they would much rather express themselves more freely in prose. Fortunately, work in **natural language processing** has made major advances in recent years, making it increasingly possible to mine the textual notes of EHRs to identify information that might bear on the CDS process (see Chap. 8).

22.4.2 Decision-Making Process

When building CDS systems, most of the work is concentrated on the development of the reasoning system and the specification of the knowledge on which that reasoning system operates. There is a wide range of strategies, each addressing different requirements that workers in biomedical informatics have adopted when building such computational resources.

22.4.2.1 Infobuttons

The simplest, and perhaps most common, form of CDS uses contextual information from an EHR to perform information retrieval from a database of information about online documents. A person viewing data in an EHR may see selectable icons (**infobuttons**) next to the names of drugs, laboratory tests, patient problems, or other elements of the patient record. Clicking on an infobutton causes the clinical information system to perform a query on the database, providing the user with one or more immediately accessible resources that can offer more information about the item in question. Alternatively, the system may automatically query one or more of those external resources and return the results of the queries for display (Cimino et al. 2002a). Clicking on an infobutton next to a drug, for example, might allow the user to access information about customary dosing, side effects, or alternative medications (see Fig. 12.8). The query that retrieves the links to the documents is tailored based on whatever is next to the infobutton icon on the screen. The query may also take into account contextual information, such as patient-related data, the activity in which the user is engaged, and the role of the user in the health care enterprise (physician, nurse, patient, and so on).

An **infobutton manager** mediates the queries between the clinical information system and the available information resources. HL7 has created a standard for "context-aware knowledge retrieval," leading to infobutton managers that have been adopted by many commercial EHR vendors. Infobutton managers need to anticipate how the clinical context might tailor the specific query performed by any given infobutton, so that the result of the query is highly precise and relevant to the situation at hand. Detailing specifically how contextual information might alter the queries performed by each infobutton type can be tedious, and requires developers to be adept at second-guessing all the reasons that might cause a user to click on a particular infobutton. Current research concentrates on the development of a Librarian Tailoring Infobutton Environment (LITE) that promises to aid the authoring of infobutton queries via "wizards" and other user-interface conveniences.

Although infobuttons are unquestionably important knowledge resources, many people would argue that they are not true CDS systems. Infobuttons retrieve relevant information for a user, but they do not explicitly address particular *decisions* that the user needs to make. The possible reasons that a user might click on an infobutton are folded into the query specification at the time that the infobutton is created; at runtime, of course, there is no way for the system to know exactly why the user selected the infobutton. Infobutton managers therefore require sophisticated query capabilities, but they do not need to reason from a clinical situation to a particular recommendation.

When the goal is to generate a situation-specific recommendation regarding diagnosis or therapy, developers need to turn to methods that can perform some kind of inference. The sophistication of the required technique is a function of the kind of inference that is necessary to render a result for the user.

22.4.2.2 Branching Logic

From a computational perspective, there is nothing simpler than encoding an algorithm directly. Numerous CDS systems have taken problem-specific flowcharts designed by clinicians and encoded them for use by a computer. Although such algorithms have been useful for the purpose of triaging patients in urgent-care situations 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 or generic 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 for clinical care (Komaroff et al. 1974). A noteworthy exception that gained enormous attention in the early 1970s was a computer program deployed in Boston at what was then the Beth Israel Hospital (Bleich 1972); it used detailed algorithmic logic to provide advice regarding the diagnosis and management of acid–base and electrolyte disorders. More recently, such branching-logic approaches have been widely adopted in the administrative information systems that third-party payers use to process requests to pre-certify payment for expensive services such as MRI studies and elective surgery.

Although flowcharts alone often are inadequate for representing the decision making required for the execution of robust clinical-practice guidelines, the algorithmic representation of clinical procedures is extremely useful for clinicians when they think about the representation of preferred clinical workflows. It is therefore common to see a branching-logic representation of clinical protocols and guidelines as one component of the complex, heterogeneous knowledge representations needed to drive sophisticated CDSS systems, such as those described later in this section when we discuss ontology-driven decision support.

22.4.2.3 Probabilistic Systems

Because computers were traditionally viewed as numerical calculating machines, people recognized by the 1960s that they could be used to compute the posterior probability of diseases based on observations of patient-specific parameters. 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. As we mentioned earlier, among the most significant experiments were those of de Dombal and associates (1972) in England, who adopted a **naïve Bayesian model** that assumed that there are no conditional dependencies among findings (i.e., a model that could make the inappropriate assumption that the presence of a finding such as fever never affects the likelihood of the presence of a finding such as chills).

Although a naïve Bayesian model may have limitations in accurately modeling a diagnostic problem, a major strength of this approach is computational efficiency. When the findings that bear on a hypothesis are assumed to be conditionally independent, then the order in which the findings are considered in the Bayesian analysis does not matter. The computer starts by considering a given finding, the prior probability of each possible diagnosis under consideration (generally the prevalence of each diagnosis in the population), and the conditional probabilities of the finding (or the absence of the finding) given each diagnosis (or the absence of the diagnosis)—the *sensitivity* and *specificity* of the finding (see the introduction of these concepts in Chap. 2). The computer then applies Bayes' rule to calculate the posterior probability of each diagnosis given the value of the finding. The computer now is poised to update the probability of each diagnosis given the value of a second finding. The prior probability for each diagnosis in this case is not the prevalence of the diagnosis in the population, however. Having applied Bayes' rule once, we have more information than we had at the start. We can treat the *posterior probability* of each diagnosis given the first finding as the *prior probability* of the diagnosis when we apply Bayes' rule a second time. When it is time to consider a third finding, the posterior probability for each diagnosis after processing the second finding serves as the prior probability for the next application of Bayes rule. The process continues until the value of each finding has been considered. This **sequential Bayes** approach was explored as early as the 1960s for the diagnosis of congenital heart disease (Gorry and Barnett 1968) and has been used in many CDS systems since.

Much of the early interest in the sequential Bayesian approach stemmed from a conviction that it simply was impractical to construct Bayesian systems in which the assumption of conditional independence was lifted: There would be too many probabilities to assess when building the system and the necessary computation could be intractable. Recent work on the use of **belief networks**, however, has demonstrated that it actually is realistic to develop more expressive Bayesian systems in which conditional dependencies are modeled explicitly—often by taking advantage of newer algorithms for concluding the posterior probabilities that are computationally efficient in most cases. (Belief networks are described in detail in Chap. 3.) Currently, most modern CDS systems that make recommendations based on probabilistic relationships use belief networks as their primary representation of the underlying clinical situation, and then "solve" the belief network at runtime to calculate the posterior probabilities of the conditions represented in the graph. The use of belief networks is popular because the formalism makes probabilistic relationships perspicuous, overcomes the assumption of conditional independence, and enables the attendant probabilities to be learned from analysis of appropriate data sets (for example, EHR data). The approach has been demonstrated in numerous diagnostic systems, from belief networks that ascertain the status of newborns from data in the neonatal ICU (Saria et al. 2010) to belief networks that offer interpretations of biomedical image data (Kahn et al. 1997).

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 (Sox et al. 1988; Weinstein and 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 Chap. 3). One class of programs for decision-analysis is designed for use by the analysts themselves; such programs are of little use to the average clinician or patient (Pauker and 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 are a whole host of **supervised learning techniques** that can determine how data are associated with hypotheses, and that consequently can be trained on EHR data to infer conclusions based on some set of input data. For example, the decision-support capabilities of the patient monitoring systems discussed in Chap. 19 often apply statistical methods to the current data stream to infer corresponding classifications to inform care providers of the patient's current state. Regression analysis or more sophisticated techniques, such as artificial neural networks and support vector machines, when applied to routinely collected patient data, have enabled investigators to develop decision aids such as the APACHE III system (Knaus et al. 1991), which offers prediction models providing prognostic information regarding patients in the ICU (see Chap. 10).

22.4.2.4 Rule-Based Approaches

Since the 1970s, workers in medical AI have been exploring the use of methods that emphasize symbolic associations rather than purely probabilistic relationships to drive decision support (Clancey and Shortliffe 1984). This work has led to the construction of **knowledge-based systems**—programs that symbolically encode-concepts derived from experts in a field in a **knowledge base** and that use that knowledge base to provide the kind of problem analysis and advice that the expert might provide. If–then production rules, such as those in MYCIN (see Fig. 22.1), often have been used to build knowledge-based systems, as have more recent approaches that encode explicit models of the application area or of the reasoning methods required (David et al. 1993; Musen 1997). The knowledge in a knowledge-based system may include probabilistic relations, such as between symptoms and underlying diseases. Typically, such relations are augmented by additional qualitative relations, such as causality and temporal relations. When a knowledge-based system is encoded using production rules, it is referred to as a **rule-based system** (Buchanan and Shortliffe 1984).

Rule-based systems provide the dominant mechanism for developers to build CDS capabilities into modern information systems. From CDS systems that interpret ECG signals to those that recommend guideline-based therapy, rules provide an extremely convenient means to encode the necessary knowledge. Rule-based systems require a formal language for encoding the rules, plus an interpreter (sometimes called an **inference engine**) that operates on the rules to generate the necessary behavior. MYCIN, for example, required the developer to encode rules in a predefined manner using the Lisp programming language (see Fig. 22.1), and had an inference engine that could interpret the rules, determine whether the rules led to conclusions that were true or false, and, if necessary, automatically evaluate other rules in the knowledge base to help determine the truth value of a given rule that might be under consideration. Although the developers of MYCIN had to construct their own syntax for encoding rules and had to program their own inference engine to evaluate the rules, there now are many open-source and proprietary "rule engines" that provide custom-tailored editors for writing rules and inference engines that can execute the rules at runtime. For example, JESS is a popular Java-based rules engine that can be licensed from Sandia National Laboratory and that currently is free for academic use; JESS is based on a rules engine programmed in C, called CLIPS, created by NASA in the 1980s. Drools is an open-source rules engine developed by the JBoss community that also has had substantial adoption.

Developers use JESS, Drools, and proprietary rules engines to create CDS systems that contain multiple rules that, as with MYCIN, chain together to generate conclusions based on a sequence of inference steps. Decision support sometimes requires multiple rules to execute at runtime, together generating a final recommendation that derives from the consequences of the rules chaining off one another.

In most installed information systems, however, rule-based decision support is much simpler and also more limited. As in the HELP system, most CDS systems have rules that generally do not chain together, but that are triggered individually, each time either there is a relevant change to the data in a patient database that should generate an alert or there is a time event that should trigger a reminder. Each rule, or MLM, examines the state of the database and generates a corresponding action, alert, or reminder that is usually sent to a particular clinician or to members of the health care team.

Arden Syntax became an international standard for MLMs endorsed by HL7 and ANSI in 1999 (see Fig. 22.3). Arden Syntax provides a standard mechanism for declaring the variables about whose values the system will perform its reasoning (values that derive from data in the clinical information system); the conditions that, if true, would predicate specific actions; and the actions that should be taken. The standard was created with the idea that the shared syntax would allow an MLM written in an idiosyncratic representation system (for example, the one adopted by HELP) to be translated into a canonical format for execution in other information systems. The hope was that the informatics community would develop whole libraries of MLMs, all written in Arden Syntax, which could operate in any clinical environment where an information system could interpret the standard format.

A significant obstacle to the sharing of MLMs is that Arden Syntax is, in fact, just a syntax. What is missing from the standard is any notion of the *semantics* of the data on which the MLMs operate. When an MLM executes, the variables that are used in the logic of the rule are bound to values that derive from the patient database of the information system in which the MLMs operate. Arden Syntax specifies that the individual database queries needed to determine the values of the variables should appear within the "curly braces" of variable definitions in the portion of the MLM known as the "data slot" (see Fig. 22.3). What a developer should include within the curly braces depends on the particular schema of the relevant patient database and mechanism for performing queries. EHR information models and the way in which elements are coded differ from system to system. Thus, all system-specific aspects of MLM integration need to be provided within the curly braces. To adapt an MLM for use in a new environment, a programmer needs to consider the variables on which the MLM operates, determine whether those variables have counterparts in the local patient database, and write an appropriate query that will execute at runtime.

The problem is compounded because there may be assumptions regarding the semantics of the variables themselves that may not be obvious to the local implementer: If the MLM refers to serum potassium, should the logic be executed if the original specimen was grossly hemolyzed?[2] If a serum potassium value is not available in the database, but there is a value for a whole-blood potassium, should the MLM be executed using that value instead?[3] If there is no serum potassium value available for today, but there is one from last night, should the logic execute using the most recent value? MLMs cannot simply be dropped from one system into another and be shared effortlessly; rather, considerable thought, analysis, and computer skill needs to go into writing the appropriate database queries that go within the curly braces to make MLMs operational.

This obstacle to sharing MLMs that are written in the Arden Syntax is known, appropriately and whimsically, as the **curly braces problem**. The lack of standards for what goes between Arden's curly braces has been a major impediment both to the sharing of MLMs and to the creation of reference libraries of clinical decision rules. HL7 recognized this difficulty early on, and developed an abstract expression language for specifying database queries known as GELLO, which was adopted as a standard in 2005 (Sordo et al. 2004). The organization also advocated the use of a specification for the canonical kinds of data that one might find in an EHR—a *virtual* EHR—so that MLMs written in terms of GELLO queries on the virtual EHR can be translated programmatically into actual queries on patient data as available at a local institution (Kawamoto et al. 2010). In 2012, HL7 approved a **draft standard for trial use** (DSTU)

[2] If the red blood cells in a specimen *hemolyze* (burst), they release potassium, which can cause an inaccurate elevation in the measured potassium value.

[3] The *serum* is the liquid that is left when the cells are removed from whole blood.

for a virtual EHR (known for historical reasons as a *virtual medial record*, or vMR; Johnson et al. 2001a). We discuss these standards further in Sect. 22.5.2. For the reasons described in that section, despite these HL7 standards, it is unlikely that the curly braces problem will be going away anytime soon.

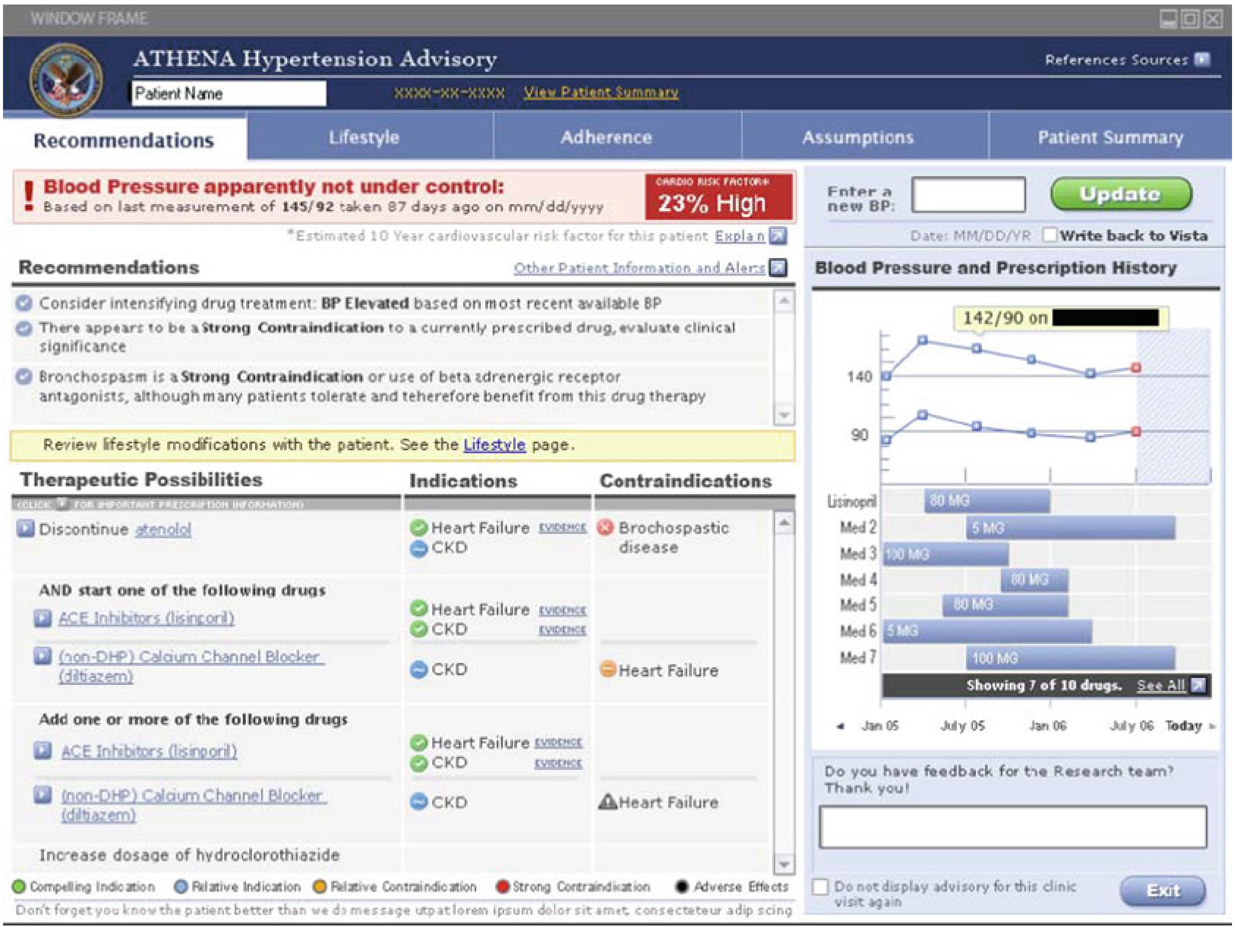
In the case of Arden Syntax, developers write rules to deal with one clinical problem at a time. There may be one MLM to deal with the problem of administering a drug like penicillin to a patient with a history of penicillin allergy; another MLM may report that a patient has a dangerously low serum potassium value. Unlike the rules in MYCIN, MLMs are generally not intended to interact with one another or to be chained together to generate complex inferences. MLMs may be coerced to chain together when one MLM posts to the patient database a value that can trigger another MLM. This mechanism also allows one MLM to set up information in the database that might invoke another MLM in the case of some future event, thus enabling the recommendation of actions that unfold over time, as in the case of many clinical practice guidelines for chronic diseases. Although this approach allows developers to program complex problem-solving behavior, the technique has the same disadvantages that came to light with chaining rule-based systems such as MYCIN: When the rule base grows to a large size, interactions among rules may have unanticipated side effects. Furthermore, when rules are added to or deleted from a previously debugged knowledge base, there may be unexpected system behaviors that emerge as a result (Clancey 1984; Heckerman and Horvitz 1986).

For MLMs to work well in practice, moreover, the rules need to be tailored to the particular clinical environment—triggered by appropriate workflow events, interacting with particular kinds of participants, customizing logic to account for various business and workflow processes, and notifying the user in setting-specific ways. To customize an MLM to account for such considerations requires that it become less portable. Much of the effort required to introduce CDS systems into the health care enterprise involves precisely such adaptations. To accelerate portability, MLM developers must seek a balance between a generic specification of logic that is widely agreed upon, and site-specific customizations that will facilitate the use of that logic. Achieving the right balance will always remain an elusive target (see also Sect. 22.5.4).

22.4.2.5 Ontology-Driven CDS Systems

There is a class of CDS systems that use higher-level abstractions of clinical knowledge and problem-solving knowledge to overcome some of the limitations of the more prevalent CDS architectures. These systems make an explicit distinction between the *static knowledge* of the clinical domain (e.g., knowledge of the specifications entailed by a clinical practice guideline) and the *problem-solving knowledge* needed to apply the static knowledge to a particular patient (e.g., the means to generate specific prescriptions for medications based on the general guideline recommendations and the particular clinical situation that the patient is experiencing). This distinction makes it possible for system builders to address different elements of the knowledge needed to be represented in the computer using tailored approaches and tools (Musen 1998; DeClerq et al. 2004).

The ATHENA-CDS system exemplifies this component-oriented approach (Goldstein et al. 2000). ATHENA-CDS is a computer system that is integrated with the HIS used by the U.S. Department of Veterans Affairs (VA), known as VistA (see Chap. 12). ATHENA-CDS is installed at several VA medical centers and has been the subject of a number of evaluation experiments (Chan et al. 2004; Lin et al. 2006). ATHENA-CDS offers advice regarding patients who have certain chronic diseases, whose physicians would like to treat those patients in accordance with recognized evidence-based clinical practice guidelines (Fig. 22.4). Currently, ATHENA-CDS draws on several electronic knowledge bases, each one capturing the knowledge of a particular guideline (e.g., for hypertension, for hyperlipidemia, for diabetes, and so on). Each time that a patient with a relevant diagnosis (e.g., hypertension) is seen in the outpatient clinic, ATHENA-CDS takes as input the corresponding guideline



Blood Pressure apparently not under control:
Based on last measurement of **145/92** taken 87 days ago on mm/dd/yyyy

CARDIO RISK FACTOR*
23% High

*Estimated 10 Year cardiovascular risk factor for this patient Explain

Enter a new BP:

Date: MM/DD/YR □ Write back to Vista

Recommendations Other Patient Information and Alerts

- Consider intensifying drug treatment: **BP Elevated** based on most recent available BP
- There appears to be a **Strong Contraindication** to a currently prescribed drug, evaluate clinical significance
- Bronchospasm is a **Strong Contraindication** or use of beta adrenergic receptor antagonists, although many patients tolerate and teherefore benefit from this drug therapy

Review lifestyle modifications with the patient. See the Lifestyle page.

Therapeutic Possibilities (CLICK FOR IMPORTANT PRESCRIPTION INFORMATION)	Indications	Contraindications
Discontinue atenolol	● Heart Failure EVIDENCE ● CKD	● Brochospastic disease
AND start one of the following drugs		
ACE Inhibitors (lisinoril)	● Heart Failure EVIDENCE ● CKD EVIDENCE	
(non-DHP) Calcium Channel Blocker (diltiazem)	● CKD	● Heart Failure
Add one or more of the following drugs		
ACE Inhibitors (lisinopril)	● Heart Failure EVIDENCE ● CKD EVIDENCE	
(non-DHP) Calcium Channel Blocker (diltiazem)	● CKD	⚠ Heart Failure
Increase dosage of hydroclorothiazide		

🟢 Compelling Indication 🔵 Relative Indication 🟠 Relative Contraindication 🔴 Strong Contraindication ● Adverse Effects

Don't forget you know the patient better than we do message utpat lorem ipsum dolor sit amet, consectateur adip scing

Blood Pressure and Prescription History

142/90 on [redacted]

Showing 7 of 10 drugs. See All

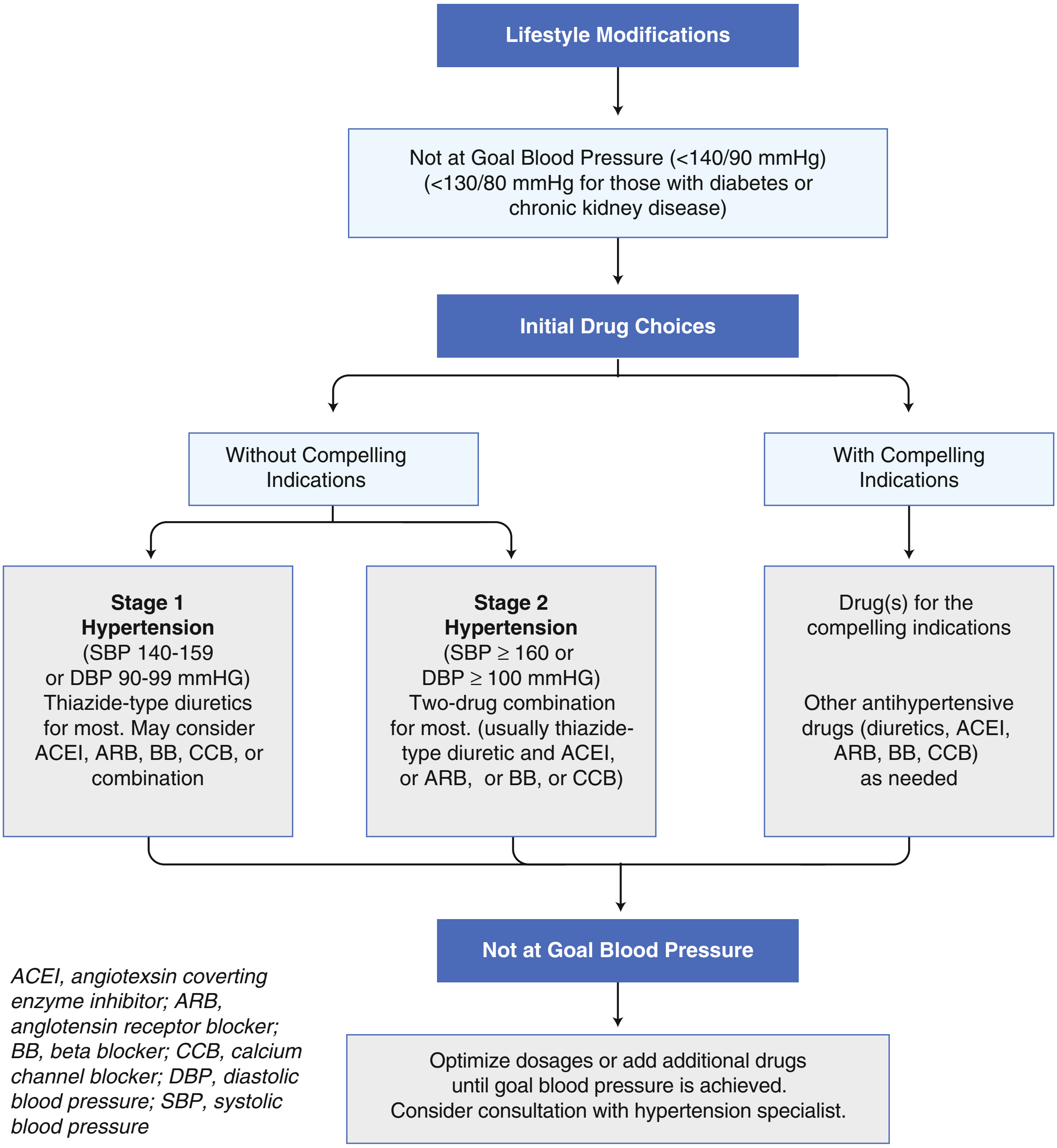
Do you have feedback for the Research team? Thank you!

□ Do not display advisory for this clinic visit again

Fig. 22.4 An example of the ATHENA-CDS system interface. ATHENA-CDS provides decision-support for the management of hypertension and several other chronic diseases by using a declarative knowledge base created as an instantiation on a generic guideline ontology. In the screen capture, the provider has entered the patient's most recent blood pressure, and is offered advice about possible alterations in therapy based on the relevant clinical-practice guideline (Source: M. K. Goldstein, VA Palo Alto Healthcare System)

knowledge base and patient-specific data from the VistA EHR, and generates as output suggestions to the clinician for treating the patient to ensure that the treatment is consistent with the care that the guideline would recommend. Because the standard documents that define clinical practice guidelines can be long and complicated, it is extremely helpful for the computer to focus the clinician's attention on precisely which interventions should be considered to guarantee that the patient's care is consonant with the medical evidence captured by a given guideline (Fig. 22.5).

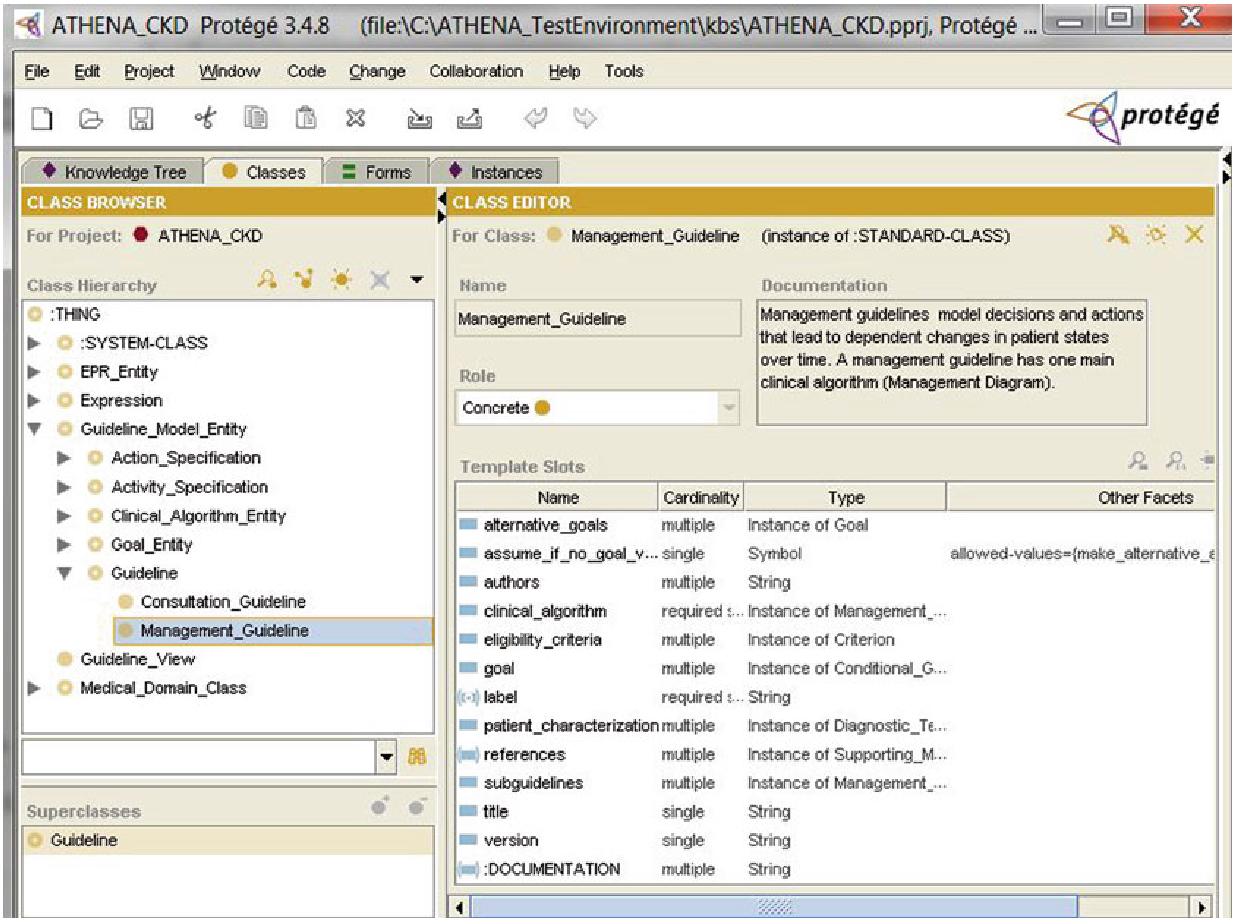
ATHENA-CDS was engineered using an approach that separates out static knowledge about the clinical application area from knowledge about problem solving (Musen et al. 1996). The developers began by creating an **ontology** of clinical guidelines in general. An ontology is like a **controlled terminology** (see Chap. 7) that includes not only an enumeration of the important entities in some application area, but also—in machine-processable form—the relationships among those entities and, possibly, constraints on those entities. Thus, an ontology contains taxonomic relationships that indicate, for example, that *cholesterol* is a kind of *lipid* or that a *serum potassium* is a kind of *laboratory test*. An ontology may also contain partitive relationships that indicate, for example, that a *systolic blood pressure measurement* is part of a *blood pressure measurement*, or that the *guideline drugs* are part of a *guideline*. To construct ATHENA-CDS, it was necessary first to define an ontology of


Lifestyle Modifications
Not at Goal Blood Pressure (<140/90 mmHg) (<130/80 mmHg for those with diabetes or chronic kidney disease)
Initial Drug Choices
Without Compelling Indications
With Compelling Indications
Stage 1 Hypertension (SBP 140-159 or DBP 90-99 mmHG) Thiazide-type diuretics for most. May consider ACEI, ARB, BB, CCB, or combination
Stage 2 Hypertension (SBP ≥ 160 or DBP ≥ 100 mmHG) Two-drug combination for most. (usually thiazide-type diuretic and ACEI, or ARB, or BB, or CCB)
Drug(s) for the compelling indications
Other antihypertensive drugs (diuretics, ACEI, ARB, BB, CCB) as needed
Not at Goal Blood Pressure
Optimize dosages or add additional drugs until goal blood pressure is achieved. Consider consultation with hypertension specialist.

ACEI, angiotexsin coverting enzyme inhibitor; ARB, anglotensin receptor blocker; BB, beta blocker; CCB, calcium channel blocker; DBP, diastolic blood pressure; SBP, systolic blood pressure

Fig. 22.5 Professional societies, health care practices, private foundations, and other organizations are all working to capture "best practices" for managing patients in accordance with scientific evidence in terms of clinical practice guidelines. Unfortunately, nearly all these guidelines are published initially as large paper documents. Here is a high-level, paper-based flowchart from the guideline developed by Joint National Commission on Hypertension. The flowchart summarizes detailed recommendations that the guideline document specifies in many pages of text. Synopses of such guidelines are available online at http://guidelines.gov

clinical practice guidelines (Fig. 22.6). The guideline ontology makes it clear that all guidelines must include *eligibility criteria* that indicate which patients should be treated in accordance with the guideline, a *clinical algorithm* that specifies the sequence of treatments recommended by the guideline, and *guideline drugs* that represent all the medications that patients might be given when their provider follows the guideline. Because the guideline ontology is general, it does not contain information about any *particular* clinical algorithm, any *particular* eligibility


ATHENA_CKD Protégé 3.4.8 (file:\C:\ATHENA_TestEnvironment\kbs\ATHENA_CKD.pprj, Protégé ...
File Edit Project Window Code Change Collaboration Help Tools
protégé
Knowledge Tree
Classes
Forms
Instances
CLASS BROWSER
CLASS EDITOR

For Project: ● ATHENA_CKD

Class Hierarchy

- :THING
  - :SYSTEM-CLASS
  - EPR_Entity
  - Expression
  - Guideline_Model_Entity
    - Action_Specification
    - Activity_Specification
    - Clinical_Algorithm_Entity
    - Goal_Entity
    - Guideline
      - Consultation_Guideline
      - Management_Guideline
  - Guideline_View
  - Medical_Domain_Class

Superclasses

Guideline

For Class: Management_Guideline (instance of :STANDARD-CLASS)

Name: Management_Guideline

Role: Concrete

Documentation: Management guidelines model decisions and actions that lead to dependent changes in patient states over time. A management guideline has one main clinical algorithm (Management Diagram).

Template Slots

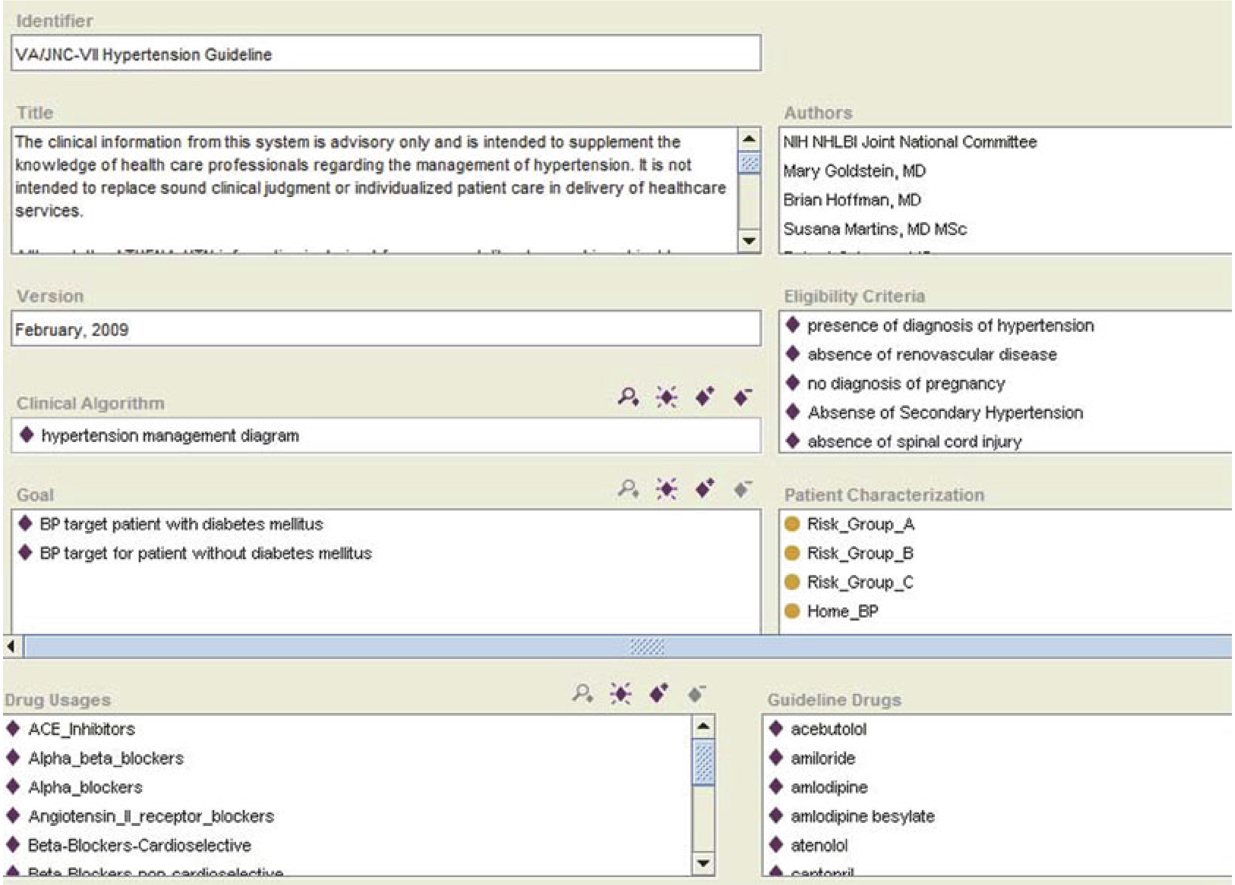
Name	Cardinality	Type	Other Facets
alternative_goals	multiple	Instance of Goal	
assume_if_no_goal_v...	single	Symbol	allowed-values={make_alternative_ε
authors	multiple	String	
clinical_algorithm	required s...	Instance of Management_...	
eligibility_criteria	multiple	Instance of Criterion	
goal	multiple	Instance of Conditional_G...	
label	required s...	String	
patient_characterization	multiple	Instance of Diagnostic_Te...	
references	multiple	Instance of Supporting_M...	
subguidelines	multiple	Instance of Management_...	
title	single	String	
version	single	String	
:DOCUMENTATION	multiple	String	

Fig. 22.6 A small portion of the ontology of clinical guidelines used by ATHENA-CDS as entered into the Protégé ontology-editing system. The hierarchy of entries on the left includes entities that constitute building blocks for constructing guideline descriptions. The panel on the right shows the attributes of whatever entity is highlighted on the left. Here, *goal*, *eligibility_criteria*, and *clinical_algorithm*, for example, are attributes of the entity known as *Mangement_Guideline*. The ontology entered into Protégé reflects concepts believed to be common to all guidelines, but does not include specifications for any guidelines in particular. The complete domain model is used to generate automatically a graphical knowledge-acquisition tool, such as the one shown in Fig. 22.7

criteria, and so on. The ontology merely states that all guidelines for management of chronic diseases have such characteristics.

Developers of ATHENA-CDS used the Protégé ontology-development system (Gennari et al. 2003) to create the ontology of clinical practice guidelines (see Fig. 22.6). They then used Protégé to allow the guideline ontology to structure **knowledge bases** that define how to manage patients in accordance with particular guidelines. The developers created a knowledge base for management of hypertension reflecting the guideline that is used by the VA and the Department of Defense (DOD), supplemented with recommendations from the Joint National Commission on Hypertension (National High Blood Pressure Education Program 2004; Fig. 22.7). They instantiated the ATHENA-CDS guideline ontology to build a knowledge base for management of congestive heart failure based on the guideline developed by the American Heart Association and the American College of Cardiology. The developers built a knowledge base for management of chronic pain, based on the guideline promoted by the VA and the DOD (Trafton et al. 2010). Other knowledge bases for guideline-based care of diabetes, hyperlipidemia, and chronic kidney disease were created in a similar manner.

The ATHENA-CDS guideline ontology can be viewed as a hierarchy of classes in an object-oriented language. Each object defines an entity in the ontology (e.g., clinical algorithm, guideline drugs). To create a knowledge base (such as the



Identifier

VA/JNC-VII Hypertension Guideline

Title

The clinical information from this system is advisory only and is intended to supplement the knowledge of health care professionals regarding the management of hypertension. It is not intended to replace sound clinical judgment or individualized patient care in delivery of healthcare services.

Authors

NIH NHLBI Joint National Committee

Mary Goldstein, MD

Brian Hoffman, MD

Susana Martins, MD MSc

Version

February, 2009

Eligibility Criteria

◆ presence of diagnosis of hypertension

◆ absence of renovascular disease

◆ no diagnosis of pregnancy

◆ Absense of Secondary Hypertension

◆ absence of spinal cord injury

Clinical Algorithm

◆ hypertension management diagram

Goal

◆ BP target patient with diabetes mellitus

◆ BP target for patient without diabetes mellitus

Patient Characterization

● Risk_Group_A

● Risk_Group_B

● Risk_Group_C

● Home_BP

Drug Usages

◆ ACE_Inhibitors

◆ Alpha_beta_blockers

◆ Alpha_blockers

◆ Angiotensin_II_receptor_blockers

◆ Beta-Blockers-Cardioselective

◆ Beta-Blockers-non-cardioselective

Guideline Drugs

◆ acebutolol

◆ amiloride

◆ amlodipine

◆ amlodipine besylate

◆ atenolol

◆ captopril

Fig. 22.7 A screen from a Protégé-generated knowledge-acquisition tool for entry of clinical-practice guidelines. The tool is generated automatically from domain ontology, shown in Fig. 22.6. The entries into the tool specify the knowledge required to treat patients in accordance with the guideline for chronic hypertension adopted by the Department of Veterans Affairs

one for hypertension management), the classes in the object hierarchy are instantiated to define the particular *clinical algorithm* mandated by the hypertension guideline, the particular *guideline drugs* required to treat hypertension, and so on. Similarly, creating the diabetes knowledge base for ATHENA-CDS required instantiating the *clinical algorithm* class with information about the sequence of events that take place in the management of diabetes, and so on. Although creating the ATHENA-CDS guideline ontology required careful analysis and modeling, it is a much easier task to use a knowledge-editing system such as Protégé to instantiate the ontology with the information required for individual guidelines. Indeed, editing and maintaining the ATHENA-CDS knowledge bases has been something that trained clinicians generally have done on their own without much assistance from workers in informatics.

In the approach demonstrated by ATHENA-CDS, software engineers create specialized computer programs that encode the procedures needed to perform different tasks using the knowledge base. For example, in ATHENA-CDS, a problem-solving program uses a guideline knowledge base (for example, the knowledge base that encodes the VA/DOD/JNC guideline for management of hypertension) in conjunction with data that the program queries from the VistA EHR to make situation-specific recommendations to providers regarding therapy for high blood pressure. The approach separates the static knowledge base from problem solvers that operate on that knowledge base. Thus, a different problem-solving program could use the knowledge base and available EHR data to determine whether a patient is eligible for treatment in accordance with the guideline. Another problem-solving program could perform **quality**

assurance to assess whether past patients have been treated in accordance with the guideline, when appropriate (Advani et al. 2003). Another program could estimate the cost of treating a patient in accordance with the guideline.

The ontology-driven approach makes it possible to start with a particular ontology (in this case, one for clinical practice guidelines for management of chronic disease) to create multiple knowledge bases, each one instantiating the ontology to specify the knowledge required for particular guidelines. Similarly, the different knowledge bases can be mapped to different problem-solving programs, such that each problem solver automates a different task associated with guideline-based care (therapy planning, eligibility determination, and so on). The ability to "mix and match" knowledge bases and problem solvers offers considerable flexibility, and enables developers to reuse elements of previous solutions to address new CDS problems that require different domain knowledge or different problem-solving procedures.

22.5 Translating CDS to the Clinical Enterprise

So far in this chapter, you have learned about the foundational elements of CDS systems. We have emphasized the research challenges that confront the informatics community to offer more useful CDS and to make it easier for developers to encode clinical knowledge in electronic form. Bringing CDS technology to the point of care, however, requires a parallel set of challenges that concern integration of CDS systems both within the information infrastructure available in real-world settings and within the workflows of their users. Deployment of CDS systems within the clinical enterprise requires an understanding of the complexities of existing HIT and of the processes by which the HIT vendor community adopts the standards that enable interoperation among different HIT components.

Over the past several decades, advanced CDS systems have been developed and deployed in a number of academic medical centers. The technology slowly has diffused into commercial EHR systems and into routine practice (Chaudhry et al. 2006). The uptake has been greater in medium-to-large hospitals and in medical-center–based networks, including affiliated practices, and has been much less in smaller hospitals, clinics, and independent practices. Although these trends have been sluggish, the recent advent of "meaningful use" regulations for HIT promises to accelerate the adoption of CDS technology dramatically.

In general, the CDS systems deployed to date in vendor EHR systems are relatively limited in scope. The greatest uptake has been in the form of simple alerts and reminders, standard physician order sets, CPOE-based prescription templates with dose checks, allergy checks, identification of drug–lab and drug–drug interactions, and some use of infobuttons or access to context-specific knowledge resources. In some specific settings, rule-based systems have been used to drive the intelligent collection of clinical information in a comprehensive, structured form (Schnipper et al. 2008b).

A few vendors have been successful at distributing knowledge resources, making available clinical knowledge in the form of drug-interaction databases, order sets for common indications, rule-based knowledge, documentation templates, and information resources for infobutton-based queries.

Nonetheless, even these kinds of resources have had relatively limited adoption to date. There are several reasons for the slow uptake of CDS, as enumerated in the sections that follow.

22.5.1 Lack of a Standard Patient Information Model

CDS rules and other knowledge forms need to operate on specific patient data. If those data are stored in a proprietary format and with non-standard encodings, then a set of rules needs to be customized to use data in that form, or the data need to be translated to a common information model. The former has been the usual process, and, as a result, vendor EHR systems tend to have libraries of rules that operate only in their own

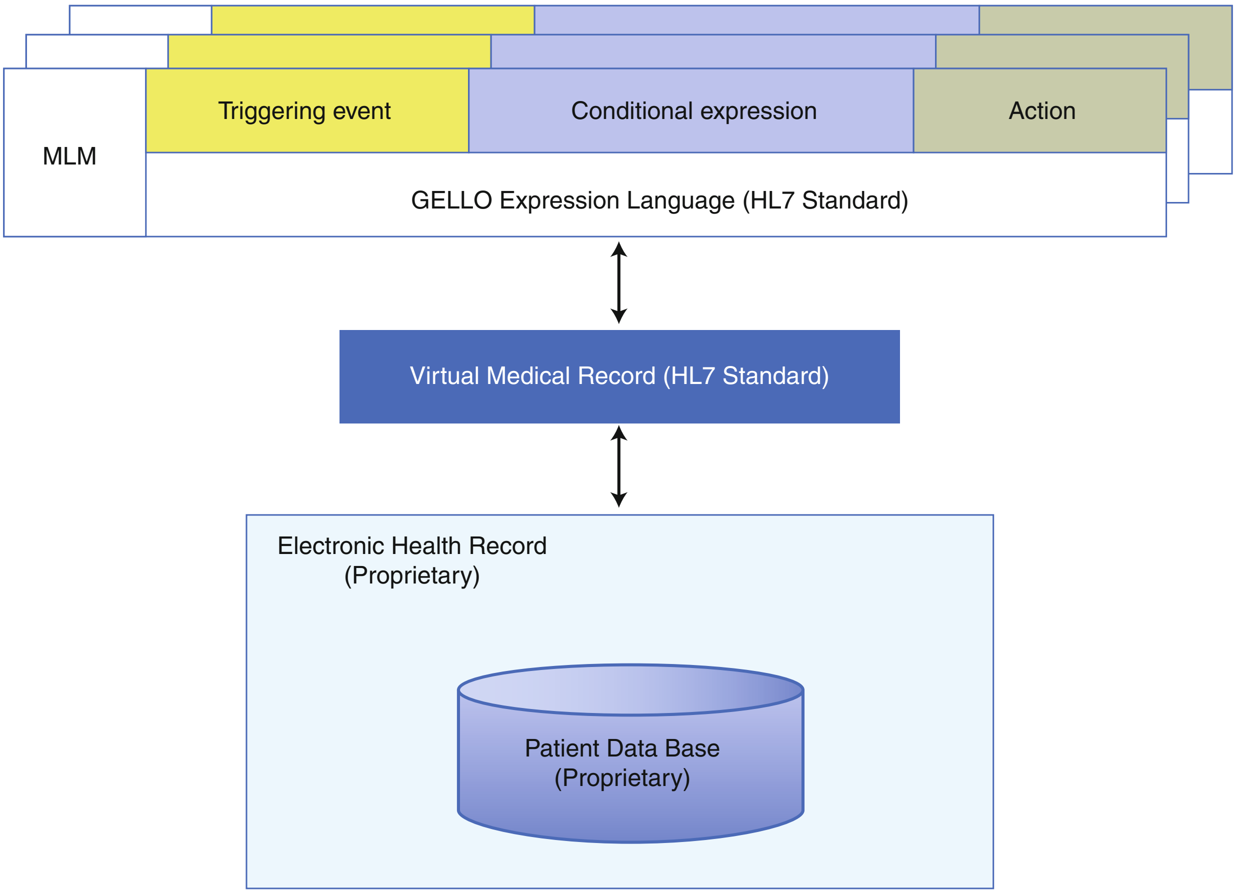


Fig. 22.8 HL7 Version 3 offers a solution to the "curly braces problem" that has to date impeded the sharing of MLMs. In Version 3, EHR vendors may adopt a standard *virtual medical record* (vMR) interface that provides a common framework (a "wrapper") for accessing patient data stored in diverse, proprietary EHR databases. In Version 3, developers of medical logic modules (MLMs) can use a standard object-oriented query language, GELLO, to access the vMR. Although it has been hoped that use of standards such as GELLO and the vMR will encourage widespread sharing of libraries of MLM decision rules, the vMR is based on the HL7 Reference Information Model (RIM), which to date has received limited adoption by the vendor community

systems, using their proprietary data dictionaries and data models; sharing across platforms and systems has been limited, and considerable work is required to integrate vendor HIT products with external CDS systems.

The informatics community is responding to this problem through the creation of new standards (see Chap. 7). For example, HL7 has developed an XML mark-up specification for the **Continuity of Care Document** (CCD), providing a standard mechanism for structuring in a static form the many data elements that are needed to record clinical encounters. Although industry adoption of the CCD specification has been slow, the use of the CCD standard offers considerable opportunity for CDS systems to interoperate across vendor platforms.

As discussed in Sect. 22.4.2, in 2012, a *virtual medical record* (vMR) based on the HL7 version 3.0 **Reference Information Model** (RIM) was approved by HL7 as a draft standard for trial use for linking dynamically at runtime the arbitrary data elements available in the patient database of an EHR to CDS systems that assume the standard vMR framework for data encoding (Kawamoto et al. 2010). The vMR thus acts as an interface between proprietary database formats and standards-based CDS systems that developers might plug into any EHR that can make its data available in a vMR-compliant manner (Fig. 22.8). HL7 is supporting ongoing work to map the vMR to standard terminologies and clinical data element definitions.

22.5.2 Lack of Adoption of Standard Knowledge Representation Models

Although Arden Syntax has been an HL7 and ANSI standard since 1999, only a few vendor systems manage their libraries of decision rules using Arden Syntax. Even doing so, of course, the rules still need to be customized for use with vendor-specific patient databases on a tedious, rule-by-rule basis to overcome the infamous curly-braces problem described in Sect. 22.4.2. As noted in that section, HL7 and ANSI adopted a query language called GELLO. GELLO can be used in rules written in Arden Syntax (and in other languages) as a standard approach for writing down the logical expressions and data queries needed for the rules to access patient-specific information stored in an EHR. GELLO assumes that the EHR data can be accessed in a manner that is consistent with the HL7 Version 3.0 RIM. GELLO itself is based on the **Object Constraint Language** (OCL) developed by the Object Management Group (OMG), and has the expressivity required both to compose complex references to clinical data elements and to construct queries. It has been hoped that GELLO will be incorporated into subsequent versions of Arden Syntax to obviate the need for curly braces. However, the HL7 Version 3 RIM has had limited uptake by vendors, who continue to rely mostly on the message-oriented HL7 Version 2 syntax for communication between systems. As a result, adoption of the GELLO query language also has been slow.

HL7 approved a standard for the Infobutton Manager in 2010 and a draft standard for order-set specification in 2012. In 2011, the organization approved a specification for a **service-oriented architecture** (SOA) to drive CDS, called the Decision Support Service (DSS).

Notably lacking from all this standards work is a shared ontology for representing clinical practice guidelines in a form suitable for execution at run time. The **Guideline Element Model** (GEM; Shiffman et al. 2000) is an XML mark-up specification that is an American National Standards Institute (ANSI) standard, now in its third revision, that guideline authors can use to annotate their narrative guidelines to identify key elements for both quality assessment and execution. GEM allows authors to demarcate the text that identifies guideline actions or eligibility criteria, and thus can serve an intermediary purpose in work to transform a prose guideline into a computable specification, but the standard does not itself provide a mechanism to translate a marked-up guideline document into a structure that a computer can interpret and execute.

A goal of some proponents is to have an ontology of clinical practice guidelines, such as the one adopted by ATHENA-CDS (see Fig. 22.6), that can inform the creation of computer-understandable knowledge bases that are able to capture knowledge about specific guidelines. Such knowledge bases then could allow a CDS system to use knowledge about the guideline, data from the EHR, and information concerning patient preferences and available resources to offer situation-specific, guideline-directed advice. An underlying infrastructure known as EON (Musen et al. 1996) drives the ATHENA-CDS system. Other ontology-based approaches have appeared over the years, including GLIF, GUIDE, PRODIGY, Pro*forma*, Asbru, and GLARE (deClerq et al. 2004). Peleg and colleagues (2003) compared many of these guideline models, and showed significant commonalities among them. Despite the large degree of agreement, however, work in this area has not yet led to anything near a standard. Part of the problem is that there is wide variation in the structure, granularity, and specificity of existing clinical practice guidelines, making it difficult to develop a single comprehensive and yet readily applicable guideline model. Analysis of the use of guidelines also indicates that guidelines themselves are rarely "executed" without considerable adaptation, except in situations such as protocol-driven care (for example, in clinical trials or in very specific procedures such as renal dialysis). ATHENA-CDS thus dispenses with offering specific guideline-based recommendations, and instead suggests to the clinicians when certain treatment options might be "compellingly indicated" or "relatively contraindicated." In highly regimented settings such as the administration of chemotherapy for cancer, however, a CDS system generally would need to be much more "prescriptive" in offering recommendations to clinicians.

22.5.3 Limited Modes of Deployment of CDS

One of the impediments to widespread adoption of CDS, particularly the use of rules and alerts, is clinician annoyance with popups, messages, emails, and other notifications that interrupt workflow. Ideally CDS systems should be integrated into the organization and presentation of information to facilitate workflow and decision making, by anticipating what information is needed for a decision, pre-fetching it, displaying it in ways that support visualization of trends or relationships, and tying these analyses to care plans or actions that can be offered immediately and quickly selected by the user. Order sets, as stated earlier, form a good example of use of CDS both to suggest appropriate actions in a given setting and to make it easy to accomplish those actions, by immediately enabling the orders in the set to be entered automatically into the EHR, perhaps with modification.

There is much ongoing research to develop methods for managing the processes of data capture, data presentation, data visualization, and selection of actions, but this work is usually being done outside of vendor EHRs. Given limited interoperability and access to the internals of proprietary systems, this kind of experimentation is now tending to take place in the form of “apps” and services that operate on externally extracted data (Mandl and Kohane 2012). Close interoperation with underlying EHRs is not currently feasible in most cases, and it remains to be seen whether the push to “apps” and services will become a force to change the industry.

22.5.4 Workflow and Setting-Specific Factors

As noted in Sect. 22.4.2, applications based on single-step situation–action rules are among the most prevalent and useful types of CDS systems. Such systems can be invoked in many contexts to provide either recommendations in real time or reminders or alerts that are processed in batch, based on time-oriented triggers or data-evaluation events. Rules can invoke other knowledge resources—providing new information content, triggering other rules, or offering order sets.

Rule content is ideally based on analysis of clinical evidence, such as recommendations or guidelines emanating from the U.S. Preventive Services Task Force, or from professional society studies of best practices for specific diseases. The job of formalizing these recommendations into executable logic requires that they be expressed in a formal way, but even having done so, such rules are not typically ready to execute in a particular environment, even if they are expressed in a rule execution language “understood” by an EHR system, and if they refer to the data elements in the EHR in its expected format. The reason they are not readily executable is also the reason that rules that work well in one environment are often not able to be successfully deployed elsewhere without substantial modification (even if in the same representation format and if using the same data model).

The reason for the failure is lack of adaptation to what we refer to as *setting-specific factors* (SSFs). To work effectively, rules need to integrate well with the clinical setting, workflow, users, application environment, and other factors. These requirements are reflected in how and when the rule should be triggered—on various events such as examination of some element of the EHR, on login to the system, or on the availability of laboratory test results. Rules may also be in developed in the form of reminders that are triggered when the CDSS evaluates on a batch basis a practice’s list of patients to be seen on a given day, the patients who have a birthday in a given month, the passage of a specific interval of time since a previous comparison event, and so on. The rules additionally may vary based on the practice setting (the emergency department, an office practice, an inpatient unit); particular inclusion or exclusion criteria or threshold modifications that may be site-specific; how the recommendation should be transmitted (electronic mail, popup windows, sidebar messages); whether the recommendation requires acknowledgment by the recipient; whether it can be overridden; whether the alert should be escalated to supervising clinicians, and so on. Rules that have been custom tailored in such ways by means of executable code naturally are less sharable than are generic rules. Failure to capture the kinds of customizations that are needed, however, makes it time consuming for individual sites to adapt generic medical recommendations to their particular requirements or to capitalize on the experiences of others. What is needed is a way to represent useful experience in terms of SSF combinations that work, without needing to do so at the level of detailed code that is difficult for users to visualize and modify.

22.5.5 Lack of a Mode for Sharing Best-Practice Knowledge for CDS

There is no established mechanism for accessing reliable, vetted libraries of best-practice knowledge in computational form that are relevant to particular clinical problem areas—for example, management of diabetes. It generally falls on each health care organization, user group, or other entity to undertake its own process of identifying and managing the best-practice knowledge it wants to deploy in its CDS systems. Even having a national or international repository of such knowledge would not preclude the need for customization, but it would certainly make it easier for each health care entity to start with a trusted source. Where such a repository should be hosted, how it might integrate public and private knowledge sources, who would have oversight over it, how knowledge would be peer reviewed and quality-rated, and how it would be sustained are among the many questions that have not yet been answered. As a consequence, health care organizations continue to perform this kind of knowledge-curation work for their own constituencies, and pilot projects often have no clear pathway to becoming operational, sustainable activities.

22.6 Future Research and Development for CDS

Workers in biomedical informatics have studied problems in assisting with complex decision making for more than half a century. It seems that it is only now, with the very recent adoption of HIT on a widespread basis, that the foundations are finally in place for the rapid advance of CDS technology in clinical settings. Although considerable logistical problems still must be surmounted as outlined in Sect. 22.5, this is an extremely exciting time in which to study CDS and its translation from the laboratory to the point of care.

22.6.1 Standards Harmonization for Knowledge Sharing and Implementation

Many implementation challenges remain for the broad adoption and effective use of CDS in EHR systems. As mentioned, one of the most active areas of current research focuses on development of standard approaches to knowledge sharing for CDS. Knowledge sharing may take the form of human-readable artifacts, machine-interpretable artifacts, or executable Web services (Osheroff et al. 2007). A capability for CDS sharing, as well as CDS functionality itself, would be substantially facilitated by the continued development and use of common standards designed to serve health care CDS needs. As noted, several standards currently exist that are aimed at specific areas of CDS and types of CDS artifacts, or that could be leveraged to benefit CDS. For example, the Clinical Decision Support Consortium, a large collaborative research and development group supported by the Agency for Healthcare Research and Quality has adopted an enhanced version of the Continuity of Care Document (CCD) to serve as the foundation for input data for multi-institutional trials of CDS technology (Middleton 2009). The CDS Consortium will soon leverage the HL7 vMR standard as the input patient information model. When taking advantage of most current standards, systems developers tend to adopt not only the standards but also particular implementation approaches, more as ad hoc solutions to specific problems than as integrated components to be used within a comprehensive framework for CDS.

To date, standards and related efforts addressing CDS overall have heavily emphasized specific

CDS execution methods and representation of the clinical context of the patient. For example, as we have noted previously, a variety of frameworks for working with rules, including Arden Syntax, Drools, JESS, and several proprietary formats, have worked their way into vendor offerings. This diversity has inhibited the exchange of best-practice knowledge. The unfortunate situation is that there are simply no repositories of clinical rules that are ready for “plug and play” adoption. Further work needs to be done to establish a common patient information model with a formal ontology, an event model for triggering events, an action model for CDS intervention recommendations, a workflow model for appropriately inserting CDS interventions into the clinical workflow, a knowledge-representation schema with a standard regular expression language, and, ideally, a measurement standard to assess CDS performance in use. In 2012, the US Office of the National Coordinator launched the Health e-Decisions initiative within its Standards and Interoperability Framework to promote coordinated community-based collaboration that would address this need. A goal of the Health eDecisions process is to create a model-driven framework for representing decision-support knowledge that can be translated among different implementation languages. As of the end of 2012, this work is being considered by HL7 as a possible standard.

22.6.2 Usability Research and CDS

The use of CDS within EHRs, and that of health IT in general, have been identified as double-edged swords: technology may provide benefit, but it also may cause considerable harm. Clinician error when using information systems that may result in untoward outcomes and unintended consequences (Karsh et al. 2010; Sittig and Singh 2009) may be an emerging property that is demonstrated only after system implementation or widespread use. Medical errors related to use of health IT are problematic, since they may represent a mismatch between the user’s model of the task being performed and the actual outcome of a computation (National Research Council (US) Committee on Engaging the Computer Science Research Community in Health Care Informatics et al. 2009), the application’s intended functionality and the resulting action or event (Harrison et al. 2007), or a latent health IT-related error yet to happen (Ash et al. 2007). Excessive alert fatigue can undermine the efficacy of clinical decision support in CPOE (Isaac et al. 2009; Strom et al. 2010), and in other IT functions (Chused et al. 2008), and result in very high user override rates (Shah et al. 2006; van der Sijs et al. 2006; Weingart et al. 2003). Critical research questions need to focus on the potential mismatch between the user’s mental model or intent and the application design and use case (Zhang and Walji 2011; Patel et al. 2010). Further attention needs to be given to basic principles of human-factors engineering, such as the use of colors and layout within the application interface. Additional questions remain regarding the ideal design of methods and controls with which a user might interact to choose a medication from a long list, or identify and encode patient problems. More advanced research will enable visualization and decision making by matching problems with care plans, and facilitation of continuity and coordination of care based on underlying CDS rules and guideline-based workflows. Especially challenging is addressing the need for structured data to support clinical decision support, and quality reporting, in a manner that is efficient for the end-user, perhaps combining structured documentation during data entry, and natural language processing for data abstraction from the clinical narrative. Most important, however, are methods to direct CDS to the right user, at the right time, with the right level of alerting.

22.6.3 Data-Driven CDS

A major area of research in informatics concerns methods for deriving knowledge from large data sets using a variety of techniques. With the adoption of health IT broadly, investigators are drawing on large-scale data-mining methods to provide CDS for population monitoring, public health surveillance, and even to offer patient-specific recommendations based on cohort data when there is no specific evidence that could otherwise guide therapy. With the increasing availability of data from diverse sources relevant to patient care, large data sets may be created and used for both discovery of previously unknown associations, and novel clinical predictions. Critical research questions here will include how to define like cohorts of patients, how to structure and frame the index decision, what methods to use to assess the likelihood of alternate prediction scenarios, and how to model and elicit the patient's preferences for each scenario. The Institute of Medicine (2011b) has articulated a long-term vision for a Learning Health System, in which clinical and administrative data of all kinds will begin to inform and enhance clinical practice on a national level in a wide variety of ways.

22.7 Conclusions

The future of CDS 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 that such innovations would 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, legislative mandates are creating not only new financial incentives but also the practical substrate of increased EHR adoption and convergence toward data interoperability, 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 that investigators understand the complex and changing nature of medical knowledge, the clearer it becomes that trained practitioners of biomedical informatics will always be required as participants in fostering a cooperative relationship between physicians and computer-based decision tools. There is no evidence that machine capabilities will ever equal the human's ability to deal with unexpected situations, to integrate visual and auditory data that reveal subtleties of a patient's problem, to work with patients to incorporate their values and priorities in care plans, 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 health care workers in the proper use of decision-support tools.

Suggested Readings

Bright, T. J., Wong, A., Dhurjati, R., Bristow, E., Bastian, L., Coeytaux, R. R., Samsa, G., Hasselblad, V., Williams, J. W., Musty, M. D., Wing, L., Kendrick, A. S., Sanders, G. D., & Lobach, D. (2012). Effect of clinical decision-support systems: A systematic review. *Annals of Internal Medicine, 157*(1), 29–43. This thorough analysis of studies of CDS systems demonstrates that there is good evidence that CDS technology can alter clinician behavior in positive ways, but that evidence that CDS systems can improve long-term patient outcomes is still inconclusive. The paper is also useful for its comprehensive bibliography.

Greenes, R. A. (Ed.). (2006). *Clinical decision support: The road ahead*. New York: Academic. This book offers a comprehensive discussion of the nature of medical knowledge and of information technology to assist with medical decision making. It provides detailed discussions of the computational, organizational, and strategic challenges in the design, development, and deployment of CDS systems.

Institute of Medicine. (2011). *Digital infrastructure for the learning health system: The foundation for continuous improvement in health and healthcare.*

Workshop series summary. Washington, DC: The National Academies Press. This monograph summarizes the vision for a national Learning Health System and offers the perspective of a wide range of thought leaders on the work required to achieve that vision.

Ledley, R., & Lusted, L. (1959). Reasoning foundations of medical diagnosis. *Science, 130*, 9–21. This is the paper that started it all. This classic article provided the first influential description of how computers might be used to assist with the process of diagnosis. The flurry of activity applying Bayesian methods to computer-assisted diagnosis in the 1960s was largely inspired by this provocative paper.

Sittig, D. F., Wright, A., Osheroff, J. A., Middelton, B., Teich, J. M., Ash, J. A., Campbell, E., & Bates, D. W. (2008). Grand challenges in clinical decision support. *Journal of Biomedical Informatics, 41*(2), 387–392. A rank-ordered list of some of the principal challenges for CDS technology development and implementation, intended "to educate and inspire researchers, developers, funders, and policy makers".

Questions for Discussion

1. Researchers in medical AI have argued that CDS systems should reason from clinical data in a way that closely matches the reasoning strategies of the very best clinical experts, as such experts are the most clever diagnosticians and the most experienced treatment specialists that there are. Other researchers maintain that expert reasoning, no matter how excellent, is at some level inherently flawed, and that CDS systems must be driven from the mining of large amounts of solid data. How do you account for the apparent difference between these views? Which view is valid? Explain your answer.
2. Transitioning CDS systems from one clinical setting to another has always been problematic. The Leeds Abdominal Pain System was installed in several major clinical settings, and yet the system never performed as well elsewhere as it had done in Leeds. The Arden Syntax, created expressly to facilitate knowledge sharing across institutions, has yet to meet this goal to a significant degree. Why kinds of setting-specific factors make it difficult to transplant decision-support technology from one environment to another? What kinds of research might lead to better methods for knowledge sharing in the future?
3. In one evaluation study, the decision-support system ONCOCIN provided advice concerning cancer therapy that was approved by experts in only 79 % of cases (Hickam et al. 1985). In another study, the HyperCritic CDS system for the management of hypertension offered the same comments that were generated by a panel of experts in only 45 % of cases (Van der Lei, et al. 1991). Such system performance is fairly typical for computer programs that suggest patient therapy. 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 problems might CDS systems encounter as their developers attempt to make the systems more comprehensive in the advice that they offer? Why might it be more difficult for computer systems to offer acceptable recommendations for patient therapy than seems to be the case for diagnosis? What safeguards, if any, would you suggest to ensure the proper use of such systems? 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 % of the time? If you would not, what level of performance would you consider adequate? Justify your answers.
4. A large international organization once proposed to establish an independent laboratory—much like

Underwriters Laboratory in the United States—that would test CDS 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 credentialing system for CDS systems, how can health-care workers feel confident in using a clinical decision aid?

5. Why did the United States federal government move to stimulate the adoption of EHRs in 2009? What mechanisms have been put in place to encourage adoption of EHRs and of CDS? What challenges remain to make CDS more pervasive in health care? Why might future clinicians be more or less attracted to CDS than is the case today?
6. There is considerable untapped potential for CDS to help in managing patients with multiple complex conditions. What are the challenges in dealing with such patients, and how can CDS be helpful? What are the features required of an algorithm that might integrate recommendations from the multiple clinical-practice guidelines that a CDS system could apply?
7. CDS is often implemented poorly, resulting in dissatisfaction, if not outright annoyance. What are the human factors that need to be taken into consideration in implementing CDS effectively? Discuss issues and approaches to enhancing usability. What are situations in which graphics and visualization might be used? How can CDS be used to enhance rather than to impede workflow? What are strategies to help avoid unintended consequences of poorly implemented CDS?