Chapter 19

Clinical Decision Support Systems

Martin Alther
Department of Computer Science
Wayne State University
Detroit, MI
martin.alther@wayne.edu

Chandan K. Reddy
Department of Computer Science
Wayne State University
Detroit, MI
reddy@cs.wayne.edu

19.1 Introduction

Clinical Decision Support Systems (CDSS) are computer systems designed to assist clinicians with patient-related decision making, such as diagnosis and treatment. Ever since the seminal *To Err Is Human* [1] was published in 2000, CDSS (along with Computer-Based Physician Order Entry systems) have become a crucial component in the evaluation and improvement of patient treatment. CDSS have shown to improve both patient outcomes and cost of care. They have demonstrated to minimize analytical errors by notifying the physician of potentially harmful drug interactions, and their diagnostic procedures have been shown to enable more accurate diagnoses. There are a wide variety of uses for CDSS in clinical practice. Some of the main uses include:

- Assisting with patient-related decision making.
- Determining optimal treatment strategies for individual patients.
- Aiding general health policies by estimating the clinical and economic outcomes of different treatment methods.
- Estimating treatment outcomes under circumstances where methods like randomized trials are either impossible or infeasible.

In 2005, Garg et al. [2] conducted a review of 100 patient studies and concluded that CDSS improved diagnosis in 64% and patient outcomes in 13% of the studies tested. That same year, Duke University conducted a systematic review of 70 different cases and concluded that decision support systems significantly improved clinical practice in 68% of all trials. The CDSS features attributed to the analysis' success included:

- natural integration with clinical workflow.
- electronic nature.
- providing decision support at the time/location of care rather than before or after the patient encounter.
- use of recommended care rather than assessments of care.

Two particular fields of healthcare where CDSS have been hugely influential are the pharmacy and billing. Pharmacies now use batch-based order checking systems that look for negative drug interactions and then report them to the corresponding patient's ordering professional. Meanwhile,

in terms of billing, CDSS have been used to examine both potential courses of treatment and conventional Medicare conditions in order to devise treatment plans that provide an optimal balance of patient care and financial expense.

In this chapter, we will provide a survey of different aspects of CDSS along with various challenges associated with their usage in clinical practice. This chapter is organized as follows: Section 19.2 provides a brief historical perspective including the current generation CDSS. Various types of CDSS will be described in Section 19.3. Decision support during care provider order entry is described in 19.4 while the diagnostic decision support is given in 19.5. Description of the human-intensive techniques that can be used to build the knowledge base is given in Section 19.6. The primary challenges with the usage of CDSS are studied in Section 19.7 while the legal and ethical issues concerned is discussed in Section 19.8. Section 19.9 concludes our discussion.

19.2 Historical Perspective

In this section, we provide a historical perspective on the development of CDSS. We will first describe the most popular early CDSS that were developed several decades ago and then we will discuss the current generation CDSS. For each of the CDSS, we will give the high-level idea of its functioning and also mention the primary drawbacks.

19.2.1 Early CDSS

Ever since the birth of the medical industry, health scientists have recognized the importance of informed clinical decision making. Unfortunately, for a long time, efficient methods for researching and evaluating such methods were quite rare. Clinicians often relied on extensive research and handwritten records to establish the necessary knowledge for a well-informed decision. Naturally, this proved to be both error prone and very time consuming. Fortunately, the evolution of business-related computing in the 1970s and 1980s gave clinicians an easy mechanism for analyzing patient data and recommending potential courses of treatment and thus, CDSS were born.

Early systems rigidly decided on a course of action, based on the user's input [3]. The user would input any necessary information, and the CDSS would output a final decision, which in turn would be the user's course of action:

- **Caduceus (aka The Internist) [4]:** This system was developed in the 1970s as a means of implementing an artificial intelligence model for use in CDSS, with the central goal of the physician using a "hypothetico-deductive" approach to medical diagnosis. One of the system's unique features was its use of a probabilistic method for ranking diagnoses. It evaluated patient symptoms and then searched its knowledge base for the most likely disease, based on the statistics of existing patients with the specified symptoms. Unfortunately, Caduceus' diagnostic accuracy was not good. For instance, in 1981, a study using pre-existing clinicopathological conference cases was conducted and then published in *The New England Journal of Medicine*. Caduceus was unable to match the diagnostic accuracy of real-life experts in this study, due to its limited knowledge base and small number of diagnostic algorithms. Thus, the system was unable to gain widespread acceptance with the medical community.

  In the mid 1980s, Caduceus evolved into **QMR (Quick Medical Reference)**. QMR differed significantly from Caduceus in that, while Caduceus was used mainly for diagnostic consultation (i.e., suggesting rigid courses of treatment to clinicians), QMR was more flexible. It allowed clinicians to modify and manipulate its suggested diagnoses/treatments in whichever

way they wished, while allowing them to utilize its knowledge base to establish their own hypotheses with regards to the treatment of more complex and difficult cases [4]. While QMR contained an extensive medical database (approximately 570 diseases in all), it had the major disadvantage of requiring frequent updates whenever new diseases were discovered. Furthermore, according to a 1994 study comparing QMR with three other clinical decision support systems, the system gave considerably fewer "correct" patient diagnoses (by the standards of a group of physicians) than the three competing systems [5]. Thus, by 2001, QMR was largely abandoned in favor of less cumbersome and more accurate CDSS.

- **MYCIN [6]:** This was originally developed in the 1970s as a means for identifying infectious diseases and recommending antibiotics for treatment. A unique aspect of MYCIN was its emphasis on artificial intelligence (AI). Its AI model was constructed through a rule-based system, in which roughly 200 decision rules (and counting) were implemented into the system, forming the knowledge base. To determine possible patient diagnoses, MYCIN's internal decision tree was consulted, and diagnostic options were reached by running through its various branches. The rule-based system was very flexible in that it allowed clinicians to either modify existing rules or devise new ones as they saw fit, making MYCIN adaptable to changing medical trends and discoveries. Therefore, it was considered an expert system, since its AI component allowed for results that were theoretically similar to those of a real-life expert.

  Unfortunately, there were many significant problems with MYCIN. First, it worked very slowly, with a typical analysis requiring upwards of 30 minutes. Second, there was concern over whether physicians ran the risk of putting too much trust in computerized results at the expense of their own judgment and inquiry. Third, there was the issue of accountability: Who would be held liable if the machine made an error in patient diagnosis? Perhaps the most important problem was how ahead of its time MYCIN was. It was developed before desktop computing and the Internet existed, so the system was based on a rather dated model for computer interaction [7]. Nonetheless, its influence was far reaching and is still felt to this day, with many systems either combining it with other expert systems (Shyster-MYCIN [8]) or using it as an influence on the development of new systems (GUIDON [9]).

- **Iliad [10]:** Iliad is another "expert" CDSS. It contains three modes of usage: Consultation, Simulation, and Simulation-Test. In *Consultation* mode, users enter real-life patient findings into the system. Iliad then analyzes these findings and compiles a list of possible diagnoses, with each diagnosis ranked in terms of its likelihood of correctness. A unique feature of Iliad is its handling of "gaps" in patient information. If the patient data appears incomplete, Iliad will suggest methods of completion and/or compromise, so that the clinician may continue working on a possible diagnosis. In *Simulation* mode, Iliad assumes the role of a complaining patient. It offers a typical real-life complaint and then demands input, testing, etc., from the clinician. The clinician's questions, responses, and diagnostic decisions are evaluated by Iliad, with feedback provided once analysis is complete. Finally, in *Simulation-Test* mode, Iliad runs a similar real-life patient simulation, except that feedback is not given to the clinician. Instead, Iliad silently evaluates his/her performance and then sends it to another user. Needless to say, because of its highly scholastic focus, Iliad is often used for educational purposes. In fact, studies have shown that it is very effective in training aspiring medical professionals for real-life practice [10].

  Unlike many other systems, which use knowledge-frame implementations, Iliad uses a framed version of the Bayes model for its analysis [11]. This makes it much easier for the system to recognize multiple diseases in a single patient (further information on Bayes classification can be found in Section 19.3.1.2). For diseases that are mutually dependent, a form of cluster analysis is included. This groups the diseases into independent categories, based not only on

the disease type, but also on clinician-specified factors such as their specific point of infection. This is so that the diseases may be efficiently analyzed and a more effective Bayesian classifier may be devised.

The 1980s saw tremendous growth and development in the field of clinical decision support. Greater involvement from the Association of American Medical Colleges in clinical library practice provided the necessary funding and resources for developing functional computerized information systems. Such systems included everything from electronic health records to financial management systems. Furthermore, PDAs (personal digital assistants) aided the development of CDSS by giving them portability. Patient data and clinical decision-making software could now be carried in the clinician's pocket, allowing him/her to easily reach informed decisions without cutting into their time with the patient. Although PDAs were more akin to basic information systems than CDSS, they were major stepping-stones in the development of CDSS that would allow clinicians to make diagnostic and treatment decisions while remaining physically close to their patients.

19.2.2 CDSS Today

Today's CDSS have much broader and more flexible methods for making clinical decisions, using both clinician and machine knowledge to give a series of potential "suggestions," with the clinician deciding on the suggestion that is most appropriate to her specific needs [3].

- **VisualDx [12]:** This is a JAVA-based clinical decision support system that, as the name suggests, is often used as a visual aid in assisting healthcare providers with diagnosis. This is useful in instances where surface level diseases (such as those of the skin) are present, and doctors need visual representations of these diseases to aid with diagnosis. A unique feature of VisualDx is that, rather than being organized by a specific diagnosis, it is organized by symptoms and other visual clues. It uses a sophisticated matching process that visually matches images of the specific patient's abnormalities with pre-existing images within a built-in database of more than 6,000 illnesses. It then uses the results of these comparisons to recommend courses of treatment.

  VisualDX has significant limitations. In addition to a vast image database, the system contains a written summary of each image. Unfortunately, these summaries are relatively brief and are, therefore, prone to overgeneralization. For example, skin biopsies are often recommended for "sicker" patients. However, it is unclear what is actually meant by "sicker." This is especially problematic when we consider that skin biopsies are rarely performed unless standard skin therapy has proven ineffective. Nevertheless, VisualDx has been demonstrated to be quite useful when diagnosing surface-level illness. The system is operational to this day, with a significant update in 2010 enabling companionship with a similar product called UpToDate [3].

- **DXplain [13]:** This is a web-based diagnosis system developed in the late 1980s by the American Medical Association. A unique feature of this system is its simplicity: Clinicians enter patient information using nothing but their own medical vocabulary, and the system outputs a list of potential diagnoses from a knowledge base consisting of thousands of diseases (with up to ten different references each), along with the potential relevance of its choices. Therefore, it functions as a clinical decision support system for physicians with little computer experience.

  DXplain has been demonstrated to be both reliable and cost efficient, especially in academic environments [3]. For example, a 2010 study consisting of more than 500 different diagnostic cases was assigned to various Massachusetts General Medicine residents. They concluded that medical charges, Medicare Part A charges, and service costs significantly decreased when

using DXplain for diagnostic recommendation [14]. DXplain has also been frequently demonstrated to give very accurate diagnoses. For example, in a 2012 study conducted by Lehigh University, the system was compared with four other CDSS. The conclusion drawn was that it was second only to Isabel (discussed below) in terms of accuracy [15].

- **Isabel [16]:** This is one of the most comprehensive CDSS available. Like DXplain, it is a web-based system designed with physician usability in mind. Originally, it focused mainly on pediatrics, but it was soon expanded to cover adult symptoms. Isabel contains two subsystems: a diagnostic checklist utility and a knowledge mobilizing utility. The diagnosis checklist tool enables physicians to enter patient demographics and clinical features into the system, which then returns a set of recommended diagnoses. The knowledge mobilizing utility may then be used to research additional information about the recommended diagnoses [3].

  Isabel has been demonstrated to give exceptionally accurate diagnoses of most patient cases. In the Lehigh University study, for example, it was shown to be the most accurate of the five systems tested. Other studies, such as a 2003 study conducted by the Imperial College School of Medicine, have also demonstrated this system to be very accurate [17]. Unfortunately, Isabel is a relatively new CDSS and, thus, more extensive testing must be performed in order to give a firm assessment of its overall reliability.

19.3 Various Types of CDSS

There are two main types of clinical decision support systems: **Knowledge-Based** and **Nonknowledge-Based.**

19.3.1 Knowledge-Based CDSS

Contemporary CDSS are rooted in early expert systems. These systems attempted to replicate the logic and reasoning of a human decision maker, reaching firm decisions based on existing knowledge. Knowledge-based CDSS rose out of the intuitive realization that medicine was a good field for applying such knowledge. A computer could (theoretically) mimic the thought processes of a real-life clinician and then give a finalized diagnosis based on the information at hand (Figure 19.1).

During the 1990s and 2000s, however, CDSS moved away from attempting to make rigorous clinical decisions in favor of offering a variety of possible diagnostic/treatment options and then allowing the clinician herself to make a finalized decision [7]. There are multiple reasons for this change in focus. These include an underlying fear of computers being inherently prone to errors, the realization that artificial intelligence still had a long way to go before it could successfully mimic the knowledge and reasoning skills of real-life clinicians, the infringement computerized decision making placed on physician/patient relations, etc. Thus, today's CDSS present a variety of diagnostic/treatment options to clinicians, allowing them to evaluate first-hand the patient's symptoms and personal testimonies while utilizing the systems as reference points for possible diagnoses.

Knowledge-based CDSS are those with a built-in reference table, containing inbred information about different diseases, treatments, etc. They use traditional AI methods (such as conditional logic) to reach decisions on courses of treatment. There are three main parts to a knowledge-based CDSS. They are the knowledge base, the inference engine, and the user communication method.

The **knowledge base** is essentially a compiled information set, with each piece of information structured in the form of IF-THEN rules. For example, IF a new order is placed for a slowly-changing blood test, AND IF the blood test was ordered within the past 48 hours, THEN we alert the

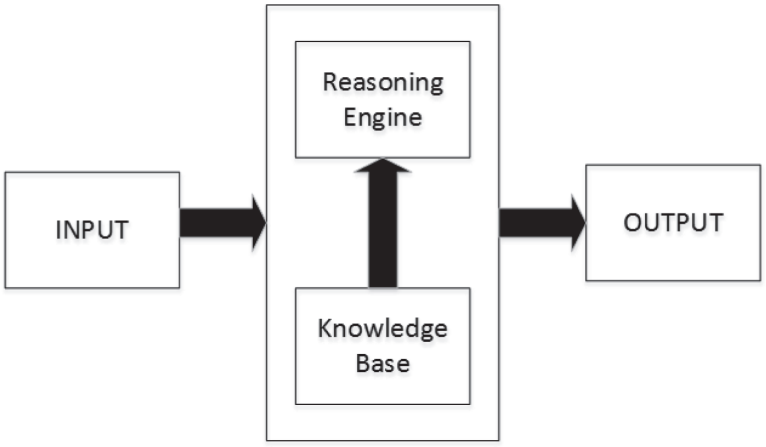


FIGURE 19.1: A general knowledge-based clinical decision support system.

physician to the possibility of duplicate test ordering. The knowledge base functions in conjunction with whichever algorithmic structure the system uses for its analysis. To put it simply, the user inputs patient information, and then the system searches through its knowledge base for matching diseases or treatment possibilities [2].

The **inference engine** applies a system of logic to the knowledge base, allowing it to "become smarter" by establishing new and/or updated knowledge. It contains the necessary formulae for combining the rules in the knowledge base with any available patient data, allowing the system to create patient-specific rules and conditions based on its knowledge of both the patient's medical history and the severity of his/her current condition. A particularly important aspect of the inference engine is its mutual exclusion from the knowledge base. Because CDSS development is very time consuming, reusability is key. Anybody should be allowed to construct a new CDSS through an existing inference engine. Unfortunately, most real-life systems are developed with a specific goal in mind (for example, diagnosing breast cancer). Thus, it is either difficult or impossible to use them beyond their intended purpose.

Finally, the **user communication method** is where the clinician herself inputs the patient's relevant data and then receives the corresponding results. In some CDSS, the patient data must be manually entered. Most of the time, however, patient data is provided through a computer-based record. The record is inputed either by the clinician or an external lab or pharmacy and is, thus, already electronically scaled. It is the clinician's job to properly manipulate the system to obtain the outcome she wishes. Diagnostic and treatment outcomes are generally represented as either recommendations or alerts. Occasionally, if an alert has been generated after an initial order was placed, automated emails and wireless notifications will be sent.

The usual format for a knowledge-based CDSS is that the clinician is asked to supply a certain amount of input, which is then processed through both the system's knowledge base and reasoning engine. It then outputs a series of possible diagnostic or treatment options for her.

19.3.1.1 Input

While there is substantial variance in the manner in which clinical information is entered into a CDSS, most systems require the user to choose keywords from his/her organization's word dictionary. The challenge clinicians typically face with this requirement is that different CDSS have different word vocabularies. The quality of output in a CDSS depends on how well its vocabulary

matches the clinician's keywords. In general, however, items related to the patient's medical history and current symptoms are going to be the suggested input.

One potentially effective method of giving detailed input is to use an explicitly defined time model, in which the user specifies various time intervals and the events that occurred within them. Unfortunately, this complicates user input and would, thus, likely prove too cumbersome for the average clinician. A simpler solution would be to use an **implicit time model**, in which broad temporal information is part of the specified user input (for example, "history of recent exposure to strep") [7]. While this simplified approach has the disadvantage of temporal ambiguity (does "recent" mean "just last week" or "last year"?), it has proven to be a viable method of measuring time in a CDSS.

19.3.1.2 Inference Engine

The inference engine is the part of the CDSS that combines user input with all other necessary data to devise a final list of "decisions." To avoid confusion, this process is usually hidden from the user. There are many different methods of analyzing user input and devising results from it. One popular method is the utilization of production rules. These are logical IF-THEN statements that, when combined, form concrete solutions to problems. MYCIN is an example of a popular CDSS that uses production rules. However, the most popular method of probabilistic estimate in an inference engine is **Bayes' Rule**, which computes the conditional probabilities [7]. In mathematical terms, suppose we wish to compute the probability of event A given event B, (or $Pr(A|B)$). As long as we already have $Pr(B|A)$, along with "prior probabilities" ($Pr(A)$ and $Pr(B)$) at our disposal, we may use Bayes' Rule to compute $Pr(A|B)$ as follows:

$$Pr(A|B) = \frac{Pr(A) \cdot Pr(B|A)}{Pr(B)} \tag{19.1}$$

To give a practical example, suppose we wish to learn the likelihood of a patient having hepatitis given that she has jaundice. (i.e., $Pr(hepatitis|jaundice)$). To compute this probability, we begin by computing a more obvious probability: $Pr(jaundice|hepatitis)$. Intuitively, this could be solved by studying an established series of patients with hepatitis and then calculating the ratio of patients with jaundice to the total number of patients. We would then plug the resultant probability into Bayes' Rule, along with the general likelihoods of hepatitis and jaundice among the total patient population ("$Pr(hepatitis)$" and "$Pr(jaundice)$," respectively). We, thus, obtain the following:

$$Pr(hepatitis|jaundice) = \frac{Pr(hepatitis) \cdot Pr(jaundice|hepatitis)}{Pr(jaundice)} \tag{19.2}$$

The result is an estimate of the patient's likelihood for having hepatitis, given the presence of jaundice.

In medicine, there is the challenge of computing the likelihood of two disjoint yet potentially related events happening simultaneously in a patient [7]. For example, suppose we wish to compute the probability of a patient having both pneumonia and an abnormal chest radiograph:

$$Pr(pneumonia + abnormal\,CXR) \tag{19.3}$$

Intuitively, it would appear that the solution is as follows:

$$Pr(pneumonia + abnormal\,CXR) = Pr(pneumonia) \cdot Pr(abnormal\,CXR) \tag{19.4}$$

Unfortunately, this formula will not work since the probabilities for pneumonia and abnormal chest radiography are typically very small. Thus, we would obtain an absurdly small probability for both occurring simultaneously, even though we know patients with pneumonia typically have abnormal chest radiographies. Fortunately, we may modify the formula to give a more accurate

prediction by multiplying the probability that a patient has pneumonia with the probability that she has an abnormal chest radiograph *given the presence of pneumonia*:

$$Pr(pneumonia + abnormal\,CXR) = Pr(pneumonia) \cdot Pr(abnormal\,CXR|pneumonia) \quad (19.5)$$

This will give us a much higher, and thus more accurate, probability estimate.

In general terms, we compute the probability of conditions "A" and "B" existing simultaneously in the following manner:

$$Pr(A+B) = Pr(A) \cdot Pr(B|A) \quad (19.6)$$

By slightly rearranging this equation, we obtain Bayes' Rule:

$$Pr(A|B) = \frac{Pr(A) \cdot Pr(B|A)}{Pr(B)} \quad (19.7)$$

A major roadblock when implementing Bayes' Rule is the possibility of a patient having multiple symptoms. Fortunately, this problem is slightly neutralized by the fact that most diseases are mutually exclusive of one another. With that said, a frame-based version of Bayes' Rule is used for taking all possible diseases into account. Illiad [11] is an example of a CDSS that successfully uses this mechanism. It uses a cluster-based framework that categorizes potential diagnoses by a common underlying thread (for example, chest pains). The logic used in these clusters is based not only on the dependencies of these possible diagnoses but also a user's understanding of how they would be categorized. For this very reason, Iliad uses Boolean statements [11]. Likewise, a Bayesian Network could be established through a series of Bayes' Rule implementations. This is essentially a graphical framework representing the cause-and-effect relationships of different events.

19.3.1.3 Knowledge Base

Naturally, for a CDSS to be successful, it must possess some form of medical knowledge. Furthermore, this knowledge must be implemented in whichever format the inference engine uses. Thus, a knowledge base must be created. The knowledge base contains all necessary medical information along with any rules or conditions necessary for analysis. For example, if the engine uses Bayes' Rule, medical knowledge must be encoded in such a manner that it allows for computation with this method of probabilistic estimates.

There are four forms of knowledge representation: logic, procedural, graph/network, and structured systems [18]. Logic is widely considered to be the most common form of knowledge representation. Medical knowledge is typically divided into two categories: declarative and procedural. **Declarative knowledge** consists of basic sentences and propositions stating hard facts, while **procedural knowledge** gives a more linear description of what actions or conclusions are feasible given the knowledge at hand. **Graph/network representation** is, as the name suggests, knowledge representation through the use of a graphical or network-based system (for example, DXPlain [13]), while **structured knowledge** is a categorized knowledge base.

Unfortunately, there is a crucial challenge in the implementation of knowledge bases that emphasize disease and treatment probability: many real-life probabilities in the clinical environment are unknown. While medical literature and consultation are certainly useful in terms of obtaining these probabilities, they often contain disparate numbers and estimates from one another, leaving the physician to guess the correct estimate. Furthermore, the probabilities of most diseases are dependent not only on specific symptoms but also on external factors such as the patient's geographic location and other demographical information. Lastly, knowledge bases must be regularly updated as new information becomes available. This is an ongoing issue with no clear solution, since many CDSS begin life as funded academic projects, for which maintenance must cease once funding has stopped.

19.3.1.4 Output

The output of a CDSS is generally in the form of a probabilistically ranked list of solutions. Generally, this list is in ASCII text format, but it may also be graphical. In some cases, factors other than probability are used in the ranking process. For example, in DXplain, diseases that are not necessarily likely but very risky when misdiagnosed are given special rank privileges. In fact, generally speaking, physicians are more interested in the least likely diagnoses than in the most likely ones, since less likely diagnoses are much easier to overlook.

19.3.2 Nonknowledge-Based CDSS

Nonknowledge-based CDSS differ from knowledge-based ones in that, rather than a user-defined knowledge base, they implement a form of artificial intelligence called *Machine Learning*. This a process by which a system, rather than consulting a precomposed encyclopedia, simply "learns" from past experiences and then implements these "lessons" into its knowledge base. There are two popular types of Nonknowledge-based CDSSs: Artificial Neural Networks and Genetic Algorithms [7].

19.3.2.1 Artificial Neural Networks

Artificial Neural Networks (ANN) simulate human thinking by evaluating and eventually learning from existing examples/occurrences [19]. An ANN consists of a series of nodes called "neurodes" (corresponding to the "neurons" in the human brain) and the weighted connections (corresponding to nerve synapses in the human brain) that unidirectionally transmit signals between them. An ANN contains three different components: input, output, and a hidden data processor. The input segment receives the data, while the output segment gives the finalized results. The data processing component, meanwhile, acts as an intermediary between the two. It processes the data and then sends the results to the output segment.

The structure of an ANN is very similar to that of a knowledge-based CDSS. However, unlike knowledge-based CDSS, ANNs do not have predefined knowledge bases. Rather, an ANN studies patterns in the patient data and then finds correlations between the patient's signs/symptoms and a possible diagnosis. Another significant difference is that knowledge-based CDSSs generally cover a much wider range of diseases than ANNs.

In order to function properly, ANNs must first be "trained." This is done by first inputing a large amount of clinical data into the neural network, analyzing it, and then hypothesizing the correct output. These educated guesses are then compared to the actual results, and the weights are adjusted accordingly, with the incorrect results being given more weight. We continue to iteratively run this process until a substantial number of correct predictions have been made.

The advantage of using ANN is that it eliminates the need for manually writing rules and seeking expert input. ANNs can also analyze and process incomplete data by inferring what the data should be, with the quality of analysis being consistently improved as more patient data is analyzed. Unfortunately, ANNs also have certain disadvantages. Due to their iterative nature, the training process is very time consuming. More importantly, the formulas/weights that result from this process are not easily read and interpreted. Therefore, with the system being unable to describe why it uses certain data the way it does, reliability is a major issue.

Nevertheless, ANNs have proven to be very successful in terms of predicting such diseases as oral cancer and myocardial infection. They have also been successfully used for the prediction of chronic diseases such as breast cancer recurrence [20] and have even shown promise in aiding the field of dentistry [21]. Thus, they are widely considered to be a viable method of clinical decision making.

19.3.2.2 Genetic Algorithms

The other key example of nonknowledge-based systems is the **Genetic Algorithm**. Genetic Algorithms are based on Charles Darwin's theories of natural selection and survival of the fittest. Just as species change in order to adapt to their environment, genetic algorithms regularly "reproduce" themselves in order to better adapt to the task at hand. As with Darwin's theory of "survival of the fittest," genetic algorithms generally begin by attempting to solve a problem through the use of randomly generated solutions [22]. The next step is to evaluate the quality (i.e., "fitness") of all the available solutions through the use of a "fitness function." The solutions are ranked by their fitness scores, with the more fit solutions having greater likelihood of "breeding" new solutions through the mutual exchange among themselves. These new solutions are evaluated similarly to their parent solutions, and the process iteratively repeats until an optimal solution is found.

Because of their more cumbersome nature, genetic algorithms have seen less use in clinical decision support than artificial neural networks. Nonetheless, they have been successfully used in fields such as chemotherapy administration and heart disease [23, 24].

19.4 Decision Support during Care Provider Order Entry

Care Provider Order Entry (CPOE) systems are decision support systems that allow clinicians to electronically input medical orders for whichever patients they are treating. Specifically, clinicians log in to a system and load their CPOE module and select the patient they are placing the order for. They write out the order and after successful review and modification, the order will be placed [25]. Here is an example of a typical care provider order entry form:

> While the CPOE's methodology depends on the clinician's specific domain, it is generally believed that allowing the physician to place an order and then providing feedback if the order is believed to be incorrect is the best way of handling care provider order entry. There are two reasons why this is preferred. One is that waiting on warning the physician of an inappropriate order until after it has been placed allows him/her to devise his/her own preferred course of action, discouraging overreliance on CDSS. The other reason is that a delay in warning the physician gives him/her the opportunity to correct any errors the system has detected. Whereas earlier warnings might underscore the errors and leave more room for mistakes.

In general, CPOE responsiveness depends on creating orders at the appropriate clinical level (i.e., the clinician's level of expertise and the user's specific condition). Unfortunately, because physicians and nurses generally have different ways of viewing these orders than the people carrying them out (pharmacists, radiologists, etc.), there tends to be confusion between the more general order of a physician and the corresponding technical terms for its content by whichever ancillary departments he/she consults. The accepted solution to this problem is for CPOE systems to avoid asking clinicians to perform tasks that fall outside their line of expertise. Pharmacists, for example, typically use pharmaceutical systems to fill and dispense whatever is specified in the CPOE system. If a higher level order is specified by the physician, the CPOE system could evaluate the pharmacy's own terminology and floor stock inventory and then determine the correct item to give the patient, giving the pharmacist more time to evaluate factors such as the order's clinical validity, safety, and efficiency [7].

Roles of Decision Support within CPOE—Decision support has several roles in CPOE [25]:

1. *Creating legible, complete, correct, rapidly actionable orders:* A CPOE system is able to avoid many of the traps/failings that often come with handwritten reports [26]. For example,

illegibility and incorrectness. Improved legibility is able to both reduce errors and reduce the amount of time clinical staff spends deciphering handwriting. Meanwhile, a "complete" order contains all necessary information to successfully place an order, while a "correct" order meets the requirements for safe and effective patient care. Needless to say, most CPOE systems are designed to ensure that both conditions are satisfied.

2. *Providing patient-specific clinical decision support:* A successful CPOE system should be able to generate decision support recommendations based on a patient's individual conditions. It should be able to generate a safety net for the clinician by merging patient-specific information (age, allergies, existing medications, etc.) with the general rules for proper practice. It should also improve patient care by promoting evidence-based clinical practice guidelines through factors such as order history or computer-based advice.

3. *Optimizing clinical care:* As the clinicians becomes accustomed to a CPOE system, they consider ways of customizing it so that their work becomes easier and more effective. Not only does this cater the system to the user's liking, but it could reduce the potential for violations such as inappropriate testing. For example, at Vanderbilt University, users of a system called WizOrder were encouraged to modify the program so that they could create Registry Orders where billing information would be more easily transferred. The challenge, in this case, comes from the need to improve the effectiveness of the system while maintaining usability. Thus, it is generally left up to the user to design a system that is able to successfully balance these two issues.

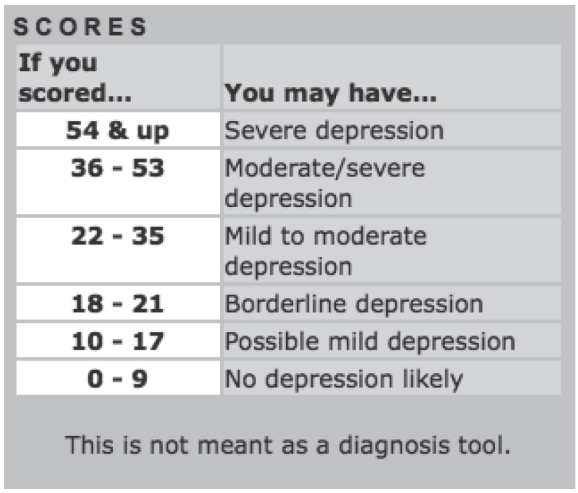
4. *Providing just-in-time focused education relevant to patient care:* Most CPOE systems provide useful educational prompts and links to more detailed description about their material, with the interface designed in a manner that encourages their use. These can be used in treatment summaries or through a corresponding web browser. Such links have the benefit of assisting the clinician with more complex orders.

Benefits and Challenges—The benefits of CPOE systems are that they can improve clinical productivity, provide solid educational support, and positively impact how patient care is given. They also make order entry much easier for both the clinician and the user, providing a computerized framework for placing orders. Thus, issues such as sloppy handwriting are nonexistent, while typos may be corrected through a built-in autocorrect feature. On the other hand, the manner in which error checking is handled may result in placing the orders containing unidentified errors. This could be especially dangerous if the order happens to be costly and critical to the patient's survival. If there is an error in it, then whatever money was spent on the order may get wasted. Worse yet, the patient's life may be in danger. Computerized order entry systems also have the disadvantage of relying on an Internet-based framework, meaning occasionally bad transmissions and server problems are inevitable.

19.5 Diagnostic Decision Support

Diagnostic Decision Support Systems are designed to "diagnose" diseases and conditions based on the parameters given as the input. In formal terms, diagnosis can be defined as "the process of determining by examination the nature and circumstances of a diseased condition [27]." What this means is that clinicians study the patient's life history before the illness has begun, how the illness came to be, and how it has affected the patient's current lifestyle [28]. Additionally, clinicians must ensure that the patient recognizes the seriousness of the disease and how to properly treat it.

Diagnostic Decision Support Systems attempt to replicate the process of diagnosis in a computerized format. The patient is asked a series of questions, and then a hypothetical diagnosis or set of possible diagnoses is output to him/her. The most user-centered systems give questionnaires inquiring about everything from the patient's family history to the patient's current health conditions. Upon completion, the patient is given a printout summarizing the conclusions drawn by the system and then suggesting possible courses of action. Similarly, there are certain medical websites sometimes offering diagnostic tools for assessing patients and recommending possible courses of treatment. A good example is Mayo Clinic's depression test [29]. It asks the patient to answer a series of questions relating to symptoms, family history, etc. (Figure 19.2) It then uses the answers to determine whether it would be a good idea to consult a professional psychiatrist for further examination.



SCORES

If you scored...	You may have...
54 & up	Severe depression
36 - 53	Moderate/severe depression
22 - 35	Mild to moderate depression
18 - 21	Borderline depression
10 - 17	Possible mild depression
0 - 9	No depression likely

This is not meant as a diagnosis tool.

FIGURE 19.2: The scoring criteria for Mayo Clinic's depression test. It explicitly states that it is not meant to be used as a diagnostic tool.

An organization known as the Foundation for Informed Medical Decision Making (FIMDM)[1] has worked to expand upon the traditional diagnostic decision support process by focusing primarily on treatment decisions that take into account the patient's personal preferences in terms of health outcomes. Specifically, they use video clips to depict the possible outcomes of each treatment, giving the patient an idea of what the experiences relating to these outcomes will be like and better preparing the patient for the clinical decision-making process. FIMDM provides tools for many diseases, ranging from breast cancer to coronary artery disease. Offline CD ROM-based software also exists for diagnostic decision support. Interestingly, in some instances, such software actually provides deeper and more detailed diagnostic information than what is available on the World Wide Web. For example, the American Medical Association has the "Family Medical Guide." This is a multilevel software package consisting of seven different modules:

1. A listing of possible diseases, disorders, and conditions.
2. A map of the human body.
3. A symptom check for the purposes of self-diagnosis and/or hypothesizing.
4. A description of the ideal body.

[1] http://www.informedmedicaldecisions.org/

5. A description of possible injuries and emergencies that require immediate attention.

6. Diagnostic imaging techniques.

7. Suggestions for how the patient's caregivers may properly care for him.

The program contains a large number of symptom flow charts, accessible through either pain-site diagrams or body system diagnosis. This is where the patient's personal inquiry comes into play: each chart contains a series of questions that require him/her to answer "Yes" or "No." Upon completion, the answers are tallied up, and a patient-specific recommendation is made based on the answers provided.

There is a considerable disagreement regarding how specific computer-generated medical advice should be provided. The common belief is that too much computerized advice will break patient/clinician relations, leading patients to self-diagnose without any formal evaluation. Fortunately, medical websites offering decision support usually give a list of options rather than a rigid diagnosis. They will usually assess patient symptoms and then devise a list of possible causes (with links for further reference), aiding him/her in deciding what condition they might have while leaving a reasonable amount of leeway to make a decision on their own. The Mayo Clinic website, for example, offers "Health Decision Guides" for a small number of diseases and conditions. These give basic information, such as the nature of the condition at hand, how it is diagnosed, and a detailed description of possible treatment options (including the pros and cons of each treatment). Each page is complemented by video clips that visually and verbally describe the condition. The purpose of such a page is not necessarily to provide the patient with a specific diagnosis but to give them more concrete background information so that they may come to a more informed conclusion with his/her clinician.

19.6 Human-Intensive Techniques

In general, there are two factors that must be considered when evaluating a clinician's ability to make a successful diagnostic or treatment-related decision: the extent of the clinician's medical knowledge and how well he/she is able to apply it to clinical problem solving [30]. Thus, when building a CDSS, one should account for the knowledge it will embody along with how it will be applied. Pragmatically, a system with little knowledge will be perceived as "dumb," while one with a limited number of knowledge-based applications will be perceived as exercising "poor judgment." Therefore, when designing a CDSS, we need a concrete methodology for implementing a knowledge base that is both extensive and reliable. This means that one will need to understand how to implement an appropriate amount of factual knowledge along with a reliable system of judgment that reaches the root of the problem and solves it, while discarding any irrelevant information.

In this section, we study a critical component of implementing a steady knowledge base, namely, the acquisition of knowledge through basic human interaction. We study how knowledge is acquired by analyzing real-life thought processes, as well as knowledge and beliefs, and then we use the results of this analysis to create a factual/judgmental knowledge base. This information is normally obtained by either physically interacting with real-life clinical "experts" or giving them direct access to a computer program that stores whatever information they can offer into its knowledge base.

There are quite a few reasons for why "expert" knowledge is valuable in clinical decision support [30]:

- **Knowledge preservation:** We wish to obtain private knowledge that would not otherwise be documented or recorded. This is so that, if the expert retires or passes away, the more esoteric but important aspects of his/her knowledge will remain within the CDSS.

- **Knowledge sharing:** Expert knowledge, once implemented into a CDSS, can be distributed among different platforms and used for external purposes such as training programs.
- **Forming a solid basis for decision aiding:** Expert knowledge may be used to create updated software that allows for better decision making.
- **Revealing the expert's underlying skills:** When an expert's knowledge is regularly used, her underlying skills and strategies are demonstrated, some of which could prove very useful in aiding decision making.

Of course, heavily emphasizing the concept of consulting "experts" for knowledge begs the question: What actually constitutes an "expert"? Furthermore, how do we distinguish an expert's knowledge and line of reasoning from those of an amateur or novice? While it is obvious that an expert will have extensive experience in her domain of expertise, equally important is her ability to build upon this knowledge and successfully adapt to changes in the environment, medical landscape, etc. In other words, a true "expert" understands the "how" rather than just the "what." This skill is, unfortunately, quite difficult to replicate with factual knowledge alone. Thus, a method must be developed that will allow a CDSS to learn and function in the exact same manner as a real-life expert.

One possible method for implementing these skills is **knowledge acquisition** (or KA). This is the process of identifying and utilizing knowledge from external sources, such as real-life experts and medical documentation, and then implementing it in such a manner that it may be evaluated and then validated by either the expert or the system itself. While biomedical literature often discusses the design of knowledge-based systems and evaluates their performance, "reproducible" methods of knowledge acquisition are usually documented elsewhere (despite being strongly correlated with the creation of a CDSS).

A knowledge base may include the relationships between potential findings and diagnoses (conceptual/factual knowledge), guidelines and algorithms for successful use of this knowledge (procedural knowledge), and a system of logic for applying these guidelines/algorithms within the underlying knowledge structure (strategic knowledge). These three knowledge branches combine to form a functional decision support system, with each form of knowledge being taken into account and then possibly expanded on.

It is possible for knowledge to be obtained through more than one expert, either through a consensus survey or by manually studying the opinions of several and then combining them into one knowledge base. Knowledge can be obtained from a variety of human experts and then translated into a form that was readable to a Decision Support System, either through the input of a human knowledge engineer or through a computer system known as a **knowledge authoring system**. This is a computer system that reads and interprets knowledge from multiple sources and then combines it into a form that is linguistically and semantically consistent. Once this is done, the scaled knowledge representation is implemented into the system as either the core knowledge base or an extension of the existing base.

19.7 Challenges of CDSS

Despite the promise CDSS holds, many physicians still choose not to use them. This is because, in spite of how much they have evolved over the last forty years, there remains many challenges in the field of clinical decision support. These correspond to factors such as machine adaptability (i.e., how capable the machine is of "learning" new medical knowledge while discarding outdated

knowledge), clarity of treatment options, and how adept the machine is at making suggestions without interfering too much with patient/clinician interaction.

19.7.1 The Grand Challenges of CDSS

In 2008, a team of researchers at various medical schools compiled a list of what they considered to be the ten "Grand Challenges" of clinical decision support [31]. The ten challenges are split into three categories:

19.7.1.1 Need to Improve the Effectiveness of CDSS

Arguably the most important challenge in the field of clinical decision support is an ongoing need to improve the effectiveness of system interventions in patient/clinician matters. This means that CDSS should act as successful intermediaries between the clinician and the patient, offering diagnostic and/or treatment suggestions that are clear and useful without being intrusive. There are five mutually inclusive methods for improving these interventions.

1. Improve the human-computer interface: The human-computer interface should be as clear and intuitive as possible. An equally important need is for the interface to be designed in such a manner that clinical workflow is left uninterrupted. In their current form, CDSS tends to give alerts that are ignored by the clinician, due to a relatively poor human-computer interface. CDSS should be able to interact with the user by either unobtrusively pointing out the issues that the clinician has overlooked or adding significant pieces of knowledge to the general workflow or decision-making process, so that valid decisions may be made even if the clinician initially overlooked important details.

2. Summarize patient-level information: It might be humanly impossible to remember every major detail of a particularly complicated patient's data. However, in any case, clinicians need to be able to recall the most important facts and conclusions about the patient. Therefore, the CDSS must be able to intelligently give a quick summary of her clinical data and then create a brief synapse of the patient's medical history, current medical conditions, physiological patterns, and current treatments [31]. The patient data summarizer must also be able to summarize all the patient data in a manner that there exists a set of indicators that are able to give "at a glance" assessments of patient status. Additionally, automatic displaying of deeper and more specific clinical decision support should be possible with better data-driven derivation of a patient's condition as well as any related data available.

3. Prioritize and filter recommendations to the user: A CDSS should be able to give information that is useful specifically to the patient at hand. This information should be evaluated and prioritized based on factors such as expected mortality (or morbidity reduction), patient preference and lifestyle, cost, the general effectiveness of the treatment (if applicable), how much it would affect the patient's own comfort or external health, how much coverage is allowed by their insurance, genetics and health history, the clinician's own success history, etc. Problems typically arise from the clinician's own time restraints and the patient's limited ability to administer a large number of medications to himself or quickly make multiple difficult lifestyle changes. The biggest of these is the need to consider conflicting decision values, determining how to prioritize them and then rank them in the corresponding order, while ensuring the number of recommendations is still manageable by the clinician (i.e., reduce "alert fatigue").

4. Combine recommendations for patients with comorbidities: A major problem with today's clinical care guidelines for conditional and medicinal management is that most of them neglect the important issue of patients (especially the elderly) having multiple comorbidities and medications. In fact, the general lack of acknowledgment for existing comorbidities and issues is cited as a major reason for the underutilization of clinical guidelines by patients [31]. For example, a clinician may be seeking to treat a newly diagnosed diabetic patient but not recognize that the patient also has, for

example, chronic obstructive pulmonary disease (COPD). Thus, the clinician's treatment suggestions for treating the patient's diabetes might significantly hinder treatment of this other condition. CDSS need to be able to take this important issue into account by weeding out the guidelines that are either redundant or intrusive of the patient's current treatment. One suggestion is for CDSS to combine the recommendations of two or more guidelines (each corresponding with both the condition at hand and any existing comorbidities) and present this combined suggestion to the clinician.

5. Use freetext information to drive clinical decision support: It is commonly believed that at least 50% of patient information is in the freetext portions of an electronic health record. This is the portion that allows the clinician to provide her own commentary on a patient's condition without the restraints of specialized questions. The information contained in freetext could prove very useful in giving more specific interventions and implementing existing patient information that would not otherwise be mentioned in the medical record.

19.7.1.2 Need to Create New CDSS Interventions

1. Prioritize CDS content development and implementation: Logically, the goal of decision support content should be to provide the most accurate and relevant information to the clinician without compromising financial cost too much. Unfortunately, successful implementation of this often takes many years. Prioritizing content implementation (i.e., interventions for improving patient safety, chronic disease management, preventive health interventions, etc.) must take various factors into account, such as the intervention's inherent value to the patient, the cost of healthcare, the reliability of the data, any difficulties that might arise in implementation, and the clinician's or patient's own views of the information's relevance [31]. While this system of data prioritizing might lead to disagreement in terms of how to properly implement future CDSS, it would significantly increase the use of the most valuable CDSS; greatly impacting the cost, safety, and quality of patient healthcare. It is very possible that, over time, this approach to prioritizing clinical decision support content may be superseded by a more refined approach.

2. Mine large clinical databases to create new CDSS: It goes without saying that there are many new patient guidelines and CDS interventions that are waiting to be developed and utilized. To establish these guidelines, we must be able to develop and test new algorithms and techniques that allow researchers to mine large data sets and expand the total knowledge base, while consequently improving CDS interventions. Similarly, a system that is able to search through scientific literature and then mine data from it to suggest potential clinical decision support guidelines would be quite useful. In other words, CDSS should be able to "learn" from large databases. Such a broad task creates quite a few challenges for the designer. In addition to the technical concerns that come with creating and implementing these algorithms, we must also address the many social and political issues associated with using such large databases. For example, as these resources cross institutional/organizational boundaries, we will need to be able to maintain patient privacy.

19.7.1.3 Disseminate Existing CDS Knowledge and Interventions

1. Disseminate best practices in CDS design, development, and implementation: Many healthcare organizations have been very successful with clinical decision support. Under scrutiny, these organizations tend to share common threads relating to issues such as design, communication, clinical practice style, and management [31]. Unfortunately, this information is usually not widely available to other organizations that are looking to adopt clinical decision support. Thus, we need to develop stronger methods for identifying and executing optimal CDS practices. A possible solution is to establish a measurement system for identifying the strength and feasibility of decision support practices. The CDS implementation process would be structured in a manner that allows information from successful users to be easily accessed and utilized by others. The establishment of methods to share successful CDS implementations and experiences would greatly benefit further research and development of CDSS.

2. Create an architecture for sharing executable CDS modules and services: The next step is to create an efficient means for sharing successful CDS modules. This could be done either remotely or through an installer. Either way, the central goal is for an electronic health record to be able to "subscribe" to these services while allowing healthcare organizations to implement their own interventions with little extra effort [31]. An important component of this challenge is to identify and standardize both the definitions and interfaces of the data required by the different CDS modules. Additionally, the architecture should have a broad enough encasing of clinical knowledge that many different inferences can be made through it. It should describe the general intervention device used (alert, order set, etc.) while still allowing for experimentation and competition. Such an implementation will help to overcome several of the barriers that come with implementing clinical decision support, as well as speed up the transition from research finding to widespread practice (a process that is estimated to take up to 17 years). Ideally, future research articles and consensus statements focused on CDSS should include a sharable CDS module in their standard format.

3. Create Internet-accessible clinical decision support repositories: The goal in this case is to create a number of Internet-accessible portals for high quality clinical decision support knowledge systems. These services should be easily downloaded, maintained, modified, installed, and used on any Certification Commission for Healthcare Information Technology (CCHIT) recognized electronic health record system [31]. Naturally, there needs to be a set of firm standards for accessibility of such a system, along with different trust levels and business models to maintain durability. The repositories should support local use of content in various healthcare organizations and also local customizations while being able to respond to system upgrades. Formalized knowledge management must be established and made available to users, so that diverse knowledge for various organizational stakeholders can be utilized. Similarly, we need to ensure that the system performs inference and guidance properly and that errors do not arise when new knowledge is implemented. This is crucial so that healthcare organizations and practitioners do not need to reinvent their own rules or interventions.

19.7.2 R.L. Engle's Critical and Non-Critical CDS Challenges

R.L. Engle, Jr., a professor at Cornell University, has extensively researched the problems that come with implementing CDSS and came up with a list of other factors that he believes contribute greatly to their lack of widespread use. He divides them into two separate categories: "Critical" and "Non-Critical" issues [32].

19.7.2.1 Non-Critical Issues

Some of the non-critical issues he describes include:

1. **The dubious reliability of computers:** Computers, like all technology, are known to occasionally falter. This, of course, poses a significant danger to clinicians attempting to use them for clinical decision support, since they could potentially fail at the most inopportune times.

2. **The overall complexity of computer systems:** Computers, let alone CDSS, usually have a learning curve. Unfortunately, many clinicians do not have the time to learn the underlying nuances of computer systems. Thus, CDSS are regarded as being inefficient and unnecessary to them.

3. **Fear of competition among clinicians regarding the effectiveness of their CDSS:** While a certain amount of competition among clinicians may be useful in continuing the progression of clinical decision support, too much of such competition could seriously hinder it and possibly result in bad relations among clinicians.

4. **The generally limited nature of these programs:** Most CDSS contain very limited knowledge bases, often specializing in one particular field of medicine. Thus, many clinicians will need to invest in several different systems (each with their own unique knowledge bases) so that they may obtain a broad range of clinical decision support options.

19.7.2.2 Critical Issues

One of the "critical" issues Engle describes with CDSS is the impossibility of developing a consistently adequate database and functional set of rules/conditions. Low accuracies in practical performance are unlikely to attract a typically busy clinician to the possibility of using a CDSS.

Another "critical" issue is the relative inaccessibility of CDSS. They are usually not implemented into large-scale information systems. While they have proven quite successful when used in limited domains (such as diagnosing a single illness), it is less clear how useful they are for broader problem domains. University of California–Berkeley professors Stuart Russell and Peter Norvig have suggested that this is largely because, unlike fields such as organic chemistry, the field of medicine lacks a general theoretical model and is consequentially fraught with massive uncertainty [33]. University of New Mexico professor George Luger and colleague William Stubblefield elaborate on this observation by suggesting five "deficiencies" in the technology behind expert systems that significantly hinder clinical decision making [34]. These deficiencies are as follows:

1. Shallow domain knowledge (i.e., a rather poor understanding of human physiology).
2. A lack of robustness and flexibility. Computer systems are unable to solve, or even recognize their inability to solve, a problem that lies outside their knowledge bases. This makes them incapable of devising effective strategies for solving problems outside their knowledge bases.
3. A marked inability to give detailed explanations of conditions and decisions.
4. Difficulties in verifying the validity of a decision.
5. An inability for such systems to learn from their own experience.

Another major shortcoming of CDSS, according to Engle, is their inability to properly work with specialized data. In order for CDSS to gain further acceptance within the medical community, they must be able to work with all kinds of data. The root of this problem can be divided into two broad categories: technical design issues and human-computer interaction [32].

19.7.3 Technical Design Issues

There are several problems that need to be taken into account in terms of the technical design.

19.7.3.1 Adding Structure to Medical Knowledge

In order to properly function, CDSS need a strong understanding of their underlying domains [7]. Bare facts do not suffice. Knowledge Representation is intended to provide CDSS with information that fleshes out the meaning behind these "facts" so that they are clear to the user. The knowledge representation scheme combines with basic domain-related facts to create the core knowledge base. Within the last 25 years, researchers have created various knowledge representation schemes, ranging from simple logic predicate lists to large network structures. The strength of a knowledge representation schema has a significant impact on both the types of problems being solved and the methodology for solving them.

19.7.3.2 Knowledge Representation Formats

In general, knowledge representation schemas fall into one of four categories [7]: logic, procedural, graph/network, or structured.

Logic-based knowledge representation was the first form of representation to gain significant mainstream appeal in the field of artificial intelligence. It is generally represented in terms of propositions, which are worldly declarations deemed to be either true or false. Once a series of propositions has been established, some of them may be combined to form sentences, which may in turn be represented by variables such as P or Q and then combined to form compound declarations such as "P and Q," "P or Q," etc. These statements must be utilized in their entirety. We cannot take a part of a statement and then use that alone to devise new declarations. Fortunately, there exists a form of logical representation that will allow us to split statements and create new declarations from these splits. This is called **First Order Logic** (or, in mathematical terms, "Predicate Calculus"). This relieves us of the limitations inherent to basic variable declarations by allowing us to have "variables within variables." This newfound sense of flexibility has proven successful in making logic-based knowledge a viable form of expert system development. In fact, the popular programming language PROLOG was designed specifically for logical research and programming.

Another form of knowledge representation is *Procedural Knowledge Representation*. This is based on the fact that logic-based representations are generally declarative, meaning they are made up of True/False statements and any questions presented are answered through standard logic inference [7]. For example, if we are diagnosing a disease like anemia associated with a value of "increased" for the Mean Cell Value (or MCV), we will need to look through all the relative logic predicates and then find those that have "increased" as a value, returning any that match what we are looking for. Procedural methods, on the other hand, give more detailed information about how the knowledge base may be used to answer a question. Rather than merely being a fact checker, it gives us a "process" to look for (i.e., "IF MCV is increased, THEN conclude pernicious anemia"). These procedural statements are provided in the form of "rules." Since MYCIN, rule-based systems have been the dominant form of expert system design within the medical industry, because of how detailed yet comprehensive they are.

Another form of knowledge representation is the *Network*. This is essentially a tree of "nodes" (representing facts, events, etc.) and the edges linking them. The flexibility of network systems has been particularly influential in the rise of Bayesian expert systems since the early 1990s. Even more significantly, the ability for networks to capture knowledge forms that would otherwise be difficult to map (like causual form) has made them viable forms of medical expert systems.

Finally, there is *Structured Representation*. This depicts knowledge in a nested, categorical manner. What makes structured representation a viable option is that it is humanly easy to read and modify. An example of a structured representation being used is in the Trial Bank Project. The Trial Bank project is a joint project between the *Annals of Internal Medicine* and *JAMA* to implement the designs and outcomes of randomized trials into structured knowledge bases. Its goal is to gradually transform text-based literature into a shared, machine-interpretable resource for evidence-adaptive CDSS [35].

19.7.3.3 Data Representation

There are many different methods of structurally representing the data [7]. Some are better suited for certain tasks than others.

The first structural data representation format to gain significant mainstream acceptance is **framing** [36]. Frames are data structures that encase the core concept that is being described along with information detailing how the concept is carried out. For example, the concept of "eating out" could be represented as:

Concept: Eating Out
Location: Restaurant
Actions: Ordering food (procedure), Paying (procedure)

Another popular form of data structuring is database management. In general, there are two types of databases that are found in the medical field: relational and object oriented. **Relational Databases** are structured much like Microsoft Excel spreadsheets. They consist of a series of "records," each containing a fixed number of fields. There is a designated "primary field" in the record structure, with the remaining fields directly related to it. The records are collected and then combined into a single table, with each row representing an individual record and each column representing the record's features. A major benefit of relational databases is their flexibility: Additional columns may be added, containing additional fields, deepening the information presented. Unfortunately, a column generally cannot hold anything more complex than a single feature. Thus, in instances where there is a need for a "record within a record," **Object-Oriented Database Management Systems (OODBMS)** are preferable. OODBMS's differ from relational systems in that they allow more complex data types to be stored in fields [37].

Structured query language (SQL) may be used to inquire about a database. Unfortunately, SQL does not allow the user to draw inferences from the data. While specific knowledge processing may not be available for databases, the ability to use higher level languages to analyze them is a major advantage.

19.7.3.4 Special Data Types

In the field of decision support, it is not enough to simply have a large medical domain. Specialized data types need to also be accounted for and addressed by the system. In addition to basic descriptions of the core features in patient diagnosis (current or past diseases, any tests performed, any drugs used, etc.), we need information about how they change and evolve over time [7]. To obtain this information, we may be required to study fields outside the core problem domain. For example, to understand the possible effects of a certain medication on a patient, we first need a basic understanding of human physiology. This is a major challenge for CDSS designers, since there exists no standardized format for depicting the research and development of a field such as physiology.

Handling dynamic knowledge bases has been a major challenge in the field of artificial intelligence since it was first devised. James F. Allen was the first person to offer a potential solution for handling such data. He suggested a format consisting of "time points" and "time intervals."[38] Unfortunately, this method is believed to be computationally infeasible when used to explain all possible cause-and-effect relationships [39]. Handling time-sensitive information requires not only a representation of instances and intervals but also a method of handling the time-sensitive concepts used by humans. Unfortunately, this is very difficult to represent through a computer. Basic concepts such as distinguishing between future and past events, recognizing different time dependencies (i.e., whether time is being measured by months, days, or years) and concurrency are mandatory for clinical decision support systems to be able to successfully determine prognoses, treatment outcomes, etc.

Ever since Allen's model was proposed, several efforts attempted to expand upon it and address these crucial issues. For example, Shahar and Musen, in 1992, proposed a model that also represents events as time intervals with beginnings and ends. However, these intervals also contained unique parameters indicating the types of events being represented with them, strengthening the causal relationships between them. Constantine F. Aliferis and his colleagues recognized that systems such as QMR, Iliad and MYCIN actually worked well in their respective domains despite lacking any sorts of temporal data models [40]. Furthermore, they argue that there exists no real evidence of implicit temporal models necessarily giving better results than explicit models. Thus, they suggested

that explicit modeling is more useful for some activities (such as prognosis) than for others (such as diagnosis). Their overall conclusion is that systems relying on dynamic clinical data require explicit temporal methodologies, while those with relatively fixed knowledge bases and greater dependency on human input are able to perform well with implicit representations.

19.7.4 Reasoning

Because CDSSs were initially designed with "artificial intelligence" in mind, early medical expert systems usually focused on imitating the decision-making processes of real-life experts. That is ironic, given that most early CDSSs such as MYCIN and Pathfinder do not "reason" the way humans do. They have no significant comprehension of human anatomy/physiology, are unable to recognize temporal concepts and (most importantly) have no ability to learn/deduce new facts [7]. However, within their narrow range of knowledge, they have been demonstrated to successfully make decisions on par with a real human expert. Unfortunately, as their domain of knowledge is broadened, performance decreases. Particularly, the ability to make inferences from "first principles" and understand the effects time may have on disease processes are crucial to building robust systems with more human-like capabilities. Fortunately, many promising methods for enabling inferential/temporal reasoning within CDSSs exist. In addition to giving the system more human-like capabilities, they have also reduced the burden of large-scale calculations in networks and have helped in handling conflicting rules within the knowledge base.

19.7.4.1 Rule-Based and Early Bayesian Systems

As previously mentioned, the simplest form of reasoning used in medical diagnostic systems is propositional logic. Logic systems, by definition, apply a system of "locality" to their reasoning. This means that, if we have a statement "if a then b" and a is known to be true, then we conclude that b is true regardless of whatever else is known to be true. While locality may be useful in instances where every fact is either absolutely true or absolutely false, the field of medicine is much more complex than that. If we add the fact that "chest pain is present in esophageal reflux" to our knowledge base, then it is no longer implied that chest pain is necessarily caused by MI. Thus, locality no longer holds true within this knowledge base.

Russell and Norvig have presented three other reasons for the failure of logic-based systems in medical diagnosis. These are laziness, theoretical ignorance, and practical ignorance [33]. Laziness is demonstrated by the system designers not putting a satisfactory amount of effort into the model. Specifically, they may fail to establish a set of conditions that is deep enough to cover every possible rule without exception. Theoretical ignorance, meanwhile, is a realization that there is no uniform theory of medicine to assist with the development of a CDSS. Thus, clinicians may omit important details or conditions due to a lack of knowledge on their part. Lastly, practical ignorance is an acknowledgment that, for any particular patient, we rarely have access to all the necessary details about her even with complete knowledge of the applicable rules.

19.7.4.2 Causal Reasoning

Causal Reasoning is the use of more esoteric domain knowledge to assist with decision making. This is a popular form of reasoning, because real-life clinicians often resort to it when solving very challenging problems. University of Southern California professor Ramesh Patil has argued that causal reasoning has many benefits. Some of which include the ability to describe disease progression, the ability to analyze disease interaction, and the ability to understand certain disease mechanisms.

One of the first medical expert systems to utilize causal reasoning was CASNET [41]. This stored its knowledge as a network of pathophysiologic states, with each knowledge component organized hierarchically in terms of discovery time. For example, signs and symptoms were at the

low end of the hierarchy, since they were among the first things to be noticed when diagnosing a disease. Connections between the hierarchy's nodes were typically represented as direct causal relationships, allowing the diseases at the highest level to represent finalized diagnoses. Reasoning is performed by running through a path in the hierarchical tree, from initial findings (i.e. patient signs and symptoms) to a final disease determination.

Two more examples of expert systems that use causal reasoning are the CHF Advisor [42] and ABEL [43] systems. The CHF Advisor uses a qualitative physiological model of its domain, with a "truth maintenance system" (TMS) securing interactions between the different parameters of the knowledge base. The TMS allows the program to determine the potential impact of changes in a variable within the model, so that they may be anticipated during diagnosis. ABEL, like CASNET, models its domain in a three-level hierarchical structure. The difference is that, in ABEL, the highest level of the hierarchy represents clinical states while the lowest level represents electrolyte stores and fluidic movement. One of the most important features of this system is its ability to determine and depict situations where a hypothesis is able to explain only a portion of a finding. This allows for greater flexibility in terms of clinical decision making, giving a less rigid representation of the cause-and-effect relationships between patient signs/symptoms and diseases.

While causal reasoning is a very effective form of reasoning, it has several disadvantages. First of all, it lacks mechanical knowledge of many important diseases, making general disease-tracking systems impossible to successfully implement (although causal knowledge may be implemented more trivially in systems with such broad domains). Another problem is the ambiguity in terms of how much detail is necessary for a "complete" system. For example, in CASNET, is three levels really enough for a robust expert system, or should there be more? Perhaps the most important problem, however, is a lack of expert-like "understanding." While causal reasoning-based expert systems may possess deep knowledge of their respective domains, they generally do not understand them the way a human clinician normally would. One final issue with causal networks is the matter of temporal relationships between findings.

19.7.4.3 Probabilistic Reasoning

Another popular form of reasoning is *Probabilistic Reasoning*. As Bayesian reasoning fell out of fashion during the 1970s, new reasoning systems known as "belief networks" began to develop. These were directed acyclic networks consisting of nodes containing both conditional and probabilistic data. These nodes had parent-child relationships (with the parent nodes "pointing" to their children). The effects the parent nodes had on their children were represented by conditional probability tables.

What made these networks popular was their ability to recognize the conditional distinctness of different findings—a task that was very difficult with Bayesian networks (particularly in large domains). This was accomplished through the use of causal relationships when designing the networks along with catch-all probability estimates (or "noise parameters"), allowing for a significant amount of leeway in terms of "correctness."

19.7.4.4 Case-Based Reasoning

Case-based systems can be described as follows: A "case" is a piece of knowledge suggesting an experience that teaches a lesson crucial to the reasoner's ability to reach his/her desired conclusion [7]. Case-based systems have two distinct components: The case itself and an index for retrieving it. Each individual case, meanwhile, has three components: The problem/situation description, the solution, and the outcome. The "problem/situation description" gives the situation and/or problem at hand. The "solution" describes the process of solving the problem. The result of a solution (success or failure) is the "outcome." Each case is accessed through the case index. To solve a problem in this fashion, we need to be able to match the current problem to a previous experience. Advocates of this approach to reasoning argue that this has the following advantages: an ability to solve more

open-ended problems, an ability to solve them quickly and nonalgorithmically, and an ability to work with complex cases.

Unfortunately, case-based reasoning has several drawbacks. In larger knowledge bases, efficient indexing is a major concern. There is much debate regarding issues such as whether high or low level features should be factored into index construction and how to design a general framework for indexes. Nevertheless, case-based reasoning has proven to be very successful in CDSS construction and continues to be used in many CDSSs to this day.

19.7.5 Human–Computer Interaction

Perhaps the most important reason for the lack of widespread use of CDSS, according to Heathfield and Wyatt, is that most of them have not been designed to address the problems real-world clinicians normally face [44]. In general, they have been used for one of two purposes: limit the number of diagnostic hypotheses (which most real-life clinicians already excel at) or assist with diagnosis and treatment advice. While systems specializing in the latter have been very well received by the medical community, they are sparse compared to those that specialize in the former. Heathfield and Wyatt argue that the relatively limited number of systems designed for assisting with diagnosistic and treatment-related advice is a major reason for the lack of mainstream attention CDSS have received. The consensus is that a CDSS must account for the clinician's own work habits. It must be accessible during patient care, simple to learn, and easy to use. It can be noted that a stand-alone CDSS requiring a significant amount of input will not be used on a regular basis, due to their cumbersome nature. At the same time, the rather narrow focus of most systems suggests that they will be needed only on rare occasions, at which the simplest solution may be to forgo them altogether in favor of other decision support methods.

There are several other problems with CDSS. The primary one is the risk of there being too much focus put on computer-related technicalities (which language to use, what kind of hardware to use, etc.) at the expense of whichever problems the user is trying to solve. Another is that system designers may use the wrong models for solving problems and miscommunicate the design issues to their users. In addition, the broad and complex nature of clinical decision support makes it very vulnerable to issues such as funding, turnover, and changing organizational structures. Successful implementation of a CDSS requires that all these matters be addressed through specific organizational policies for the creation and utilization of knowledge-based tools.

Fortunately, measures have been taken to address these concerns. Problem Knowledge Couplers (PKCs), are designed with an interface that is simple enough for people outside of the medical industry to comprehend and use. While each coupler represents a single problem, in-house tutorials are included to guide the user in properly utilizing them. Unfortunately, the use of PKCs has not yet become widespread. The main reason why is because PKCs heavily invade a clinician's work environment, and it is unclear how useful they are for a large number of patients. For example, suppose the clinician has a coupler designed for headache diagnosis and management. Because most headaches are easily diagnosed and treated, the number of patients with headaches that would require a specialized computer system is very small. Many clinicians would, thus, seriously question whether such a system is worth the bother if it will only be used on rare occasions.

Perhaps the most effective method of addressing the criticisms lobbied at CDSS is the use of *Electronic Health Record Systems* (EHRSs) and *Computer-Based Physician Order Entry Systems* (CPOESs). EHRSs solve many of the issues pertaining to CDSS by providing a standardized user interface and data model for CDSS design. Access to external data (i.e., laboratory data, pharmaceutical data, etc.) is a standard feature in EHRSs and allows CDSS designers to focus more on data access and user interaction than data input. Alerts and reminder systems are included to reduce user-related errors and oversights. However, EHRS are still in their infant stages and will likely need more development time before reaching mainstream consciousness. As it stands, there exists no EHRS that are able to successfully handle the specific vocabulary and strong ontology required for

automating complex guidelines. Until a firm set of standards for EHRSs is established, it is unlikely that support for such detailed knowledge will be implemented anytime soon. CPOESs, meanwhile, have proven very useful in hospital settings. Due to the wide range of legacy systems, implementing packaged EHRS software would be very difficult in such environments. While CPOESs may not have the sophisticated data integration of an "all-in-one" EHRS, they are functional within their limited range.

It is worth mentioning that user interface issues do not necessarily disappear with the use of electronic health record and care provider order entry systems. They simply shift focus. For example, a CPOES or EHRS designed for a drug interaction study during prescription allows this function to run in the background without the user needing to explicitly invoke or shut it down. However, without the ability to modify the parameters (for example, give a warning only when severe interactions are possible) or frequency of advice, clinicians may become hesitant to use the system. Similarly, when complex automated guidelines become feasible, system designers will need to be able to seamlessly provide information while quietly relegating the process into the background. In fact, it is very possible that the biggest challenge in terms of human-computer interaction is that many humans will not wish to take advice from a machine. At this stage, the only possible solution is to let the clinician specify a "threshold of intrusiveness" for the system that, once exceeded, will allow her to ignore it completely.

19.8 Legal and Ethical Issues

Given the sensitivity of information in the field of healthcare, it is quite natural that ethics and legality would be of great concern to both clinicians and CDSS designers. Ever since the birth of clinical decision support, numerous methods have been proposed to regulate what is and what is not allowed in this domain. These addressed issues ranging from the question of who should be allowed to use a "medical computer program" to the dangers present when physician autonomy has been violated. It has been accepted by the community that computers cannot supersede human decision makers. From an ethical standpoint, computers should not be used as a substitute for basic human decision making. Surprisingly, this viewpoint has been advocated at least as much by those who are in support of clinical decision support as it has by those who are against it, due in no small part to the fact that even those who use CDSSs must be wary of breaking individual patient relations.

19.8.1 Legal Issues

Legality is a crucial component of clinical decision support. In order for the field to prosper, there must be a grounded set of a standards for how and where it can be used. Unfortunately, there is quite a bit of ambiguity regarding how this issue should be handled in the field, because medicine and computers have legal standards that are very distinct from one another.

Liability: An important question in the field of clinical decision support is that of who should be held liable for the use, lack of use, or misuse of a computerized system to aid in clinical decision making. In the United States, service providers are legally held accountable for any injuries or fatalities sustained by their users, while other countries tend to have very different standards of accountability for injury or death. In any case, liability may be addressed in one of two ways: either through the **negligence standard** or the **strict liability standard**. These are general standards of liability in cases of injury or death. The difference between the two is that the negligence standard applies to services while the strict liability standard applies to goods or products. There is ongoing debate over whether CDSS are classified as services or goods, because they share characteristics

of both. For example, a clinical diagnosis is clearly a service. However, a CDSS is a commercially manufactured item, which could just as easily classify it as a product. To further complicate things, the increasingly wide commercial availability of CDSS begs the question of what the patient's role was in a serious or fatal incident, while a clinician may be considered "negligent" if she accepts a faulty computer diagnosis or gives an errant diagnosis of her own. The clinician may also be held liable if she is believed to have violated basic reasonable person standards. Lastly, there is the question of whether a computer program classifies as an invention or a work of art. Both possibilities raise many legal questions of their own.

19.8.2 Regulation of Decision Support Software

When medical devices were regulated through the Federal Food, Drug, and Cosmetic Act of 1938, they were defined as "instruments, apparatus, and contrivances, including their components, parts, and accessories intended: (1) for use in diagnosis, cure, mitigation, treatment, or prevention of diseases in man or other animals; or (2) to affect the structure of any function of the body of man or other animals [45]." In 1976, congress devised the Medical Device Amendments, requiring that these devices were safe and effective before being sold. While, in 1990, a new regulation was established that emphasized postmarket surveillance rather than premarket approvals [46].

The FDA regards medical software as a device, falling under one of four categories:

1. **Educational and bibliographic software:** This is software intended for use in performing clerical functions such as data storage and accounting, or educational purposes. It is not used for professional medical practice and is, thus, usually not regulated.

2. **Software components:** This is software that is inherently present in medical devices, such as X-ray systems and ventilators. It is typically regulated.

3. **Software accessories:** These are typically attached to or used with physical devices. The corresponding functions include radiation treatment planning, offline study of EEG data and statistical analysis of pulse oximetry data. Because of their widespread professional use, they are actively regulated.

4. **Standalone software:** This is software that has no relation to external medical devices. CDSS fall under this category. There is continuous debate over whether standalone software should be regulated.

19.8.3 Ethical Issues

There are three major issues of ethical concern when it comes to CDSS [47]:

- **Care standards:** This implies that we must provide the best possible treatment without deviating from our personal range of care and avoid deceiving patients. The use of CDSS provides an additional layer of concerns: Do computers help or hinder our attempts at meeting these responsibilities? Do they give us any additional responsibilities? Most importantly, does the technology ultimately improve patient care? If the answer is "Yes," then we may safely say that we have met a crucial responsibility. On the other hand, if the answer is "No," then it is clear that we should not be using this technology.

  Unfortunately, the benefits of decision support (or lack thereof) are not always apparent. In some instances, it is not even possible to reach an overall consensus without experimenting on some kind of test subject at the risk of his/her own personal well-being. The idea of error avoidance is closely related to a general standard of care. Standards constantly evolve in health

professions, because they cover the actions that are most successful in achieving specific goals. To fail to adhere to these standards is to increase error risk. Because errors and their consequences are generally regarded as harmful, the obligation to adhere to these standards is an ethical one.

Ethical standards are highly empirical in nature and are, thus, open to revision. New evidence forces frequent changes in these standards. And, to be sure, the precise content of any standard might be open to debate. The "reasonable person" standard, for example, involves truths that are often ambiguous and open to interpretation. This naturally results in major disagreement among otherwise fair and reasonable people. Similarly, a "community standard" sometimes fails to identify a proper distinction between error and success under the conditions with which it may be invoked. Therefore, it may sometimes be totally permissible to violate these standards if such action results in positive outcome and few negative consequences.

In terms of computer-assisted patient diagnosis, the issue is whether or not the use of CDSS increases the risk of error. While accurate diagnosis is usually linked to optimal treatment, this does not always happen. In some cases, patients may be properly treated despite a technically inaccurate diagnosis. While, in others, the patient may be improperly treated despite a technically correct diagnosis. Additionally, computers are capable of suggesting diagnoses that fall outside of traditional clinical contexts (such as in tests for blood-borne pathogens) [7]. In other words, a crucial ethical question we find ourselves asking is whether it is acceptable to use a CDSS in the midst of scientific ambiguity.

In cases such as these, we wish to progress technological development without risking patient treatment. One approach is to exercise "progressive caution." The idea behind this is that medical informatics is and always will be a work in progress, but users and society must ensure that we properly utilize the tools we are given in moving the field forward. We wish to ethically optimize the role of decision support while maintaining appropriate levels of scrutiny and skepticism of our current work in the field.

Ever since people first began addressing ethical issues in the field of medical informatics, it has been recognized that computers, in addition to aiding in the progression of medical science, contribute to changes in basic standards of patient care. These inevitable developments increase the likelihood that computer use will be required of clinicians. While this may seem intimidating to more cautious practitioners, it also opens the door for many exciting developments and opportunities in the field of medical practice.

- **Appropriate use/users:** Naturally, there are many ways in which a computerized decision support system might be misused. It may be used for purposes beyond that with which it was intended, or it may be used without sufficient enough training. There are many problems with decision support system misuse. First of all, a tool that is specially designed with a single purpose in mind is far less likely to work properly (if at all) when used for purposes beyond that with which it was intended. For example, one may very well perform a successful colectomy with a standard kitchen knife or slice a tomato with a scalpel. However, the likelihood of success is much greater when these tools are used for their intended purposes. Similarly, a medical computer system may be improperly used if, for example, it was designed for instructional purposes but used as an aid in clinical decision making instead. If the use of clinical decision support is to become more widespread, it is imperative that new tools and systems are properly documented and used by those who are well trained in their functionality.

  Identifying what constitutes qualification in the use of CDSS is crucial. If a novice uses such a system (especially one who is not a trained physician or nurse), she may rely too much on widely available resources such as online medical references, creating a risk for misinterpretation of the system's output. Meanwhile, if a medical professional uses a CDSS without proper

training, she may improperly use the software or may not use it to its full capacity. These concerns may be addressed through a set of strict qualifications and training requirements for the users. Unfortunately, it is unclear what these qualifications should be and how much training should be required of potential users. Furthermore, there is the fear of CDSS being relied on too heavily by users. While computers have come a long way since clinical decision support was first devised in the 1970s, they are still incapable of trumping a human being in terms of cognitive functionality and interpretation. Thus, while such systems are useful in aiding decision making, they should not substitute human decision making.

- **Professional relationships:** Lastly, a major ethical issue in clinical decision support is in the field of professional relationships. Patients often place a massive amount of trust in medical professionals—sometimes too much. Meanwhile, many physicians put too little trust in patients and their judgment. This paradigm has led to the concept of "shared decision making," which is the idea that patients and caregivers should work together to make important clinical decisions. Evidence suggests that this is the most effective form of human-to-human decision making. If a computer is to be used in aiding these decisions, it must be evaluated in the exact same way.

  There are two important ethical problems that come into play here. The first is that the computer will create a barrier between the patient and the physician. Particularly, ambiguous diagnosis (especially when the stakes are high) is a major concern among both the patient and the physician. When a computer is relied on for decision making, we run the risk of committing the "computational fallacy"—the view that a computer-instigated decision is somehow more valid or accurate than a human decision. This is a potentially dangerous view not just because it undermines the physician's skills at decision making but also the patient's.

  Some of our concern may be alleviated by withholding information about the use of a CDSS from the patient. However, this raises the second important ethical question: Should patients be given this information in the first place? The answer depends on roughly two factors: The patient's general knowledge of medicine and medical statistics, and the clinician's general understanding of patient communication etiquette. In any case, it is inappropriate to use computerized output for the sake of outsmarting patients or forcing them to agree with a professional. On the other hand, as patients themselves gain further access to decision support software, they may use it to challenge the physician's viewpoints and attempt to self-diagnose. As CDSS evolve, this will become an even greater problem, because computers will play a larger role in shared decision making. Thus, overreliance on a computer's decision becomes a major risk, and the patient himself may constitute an inappropriate user.

19.9 Conclusion

Clinical Decision Support Systems provide a great opportunity for physicians to improve both the accuracy of medical diagnosis and the reliability of medical treatment. There are numerous support systems that are currently being used in clinical practice such as DXPlain and Iliad. Each system provides a unique opportunity for clinicians to indicate and diagnose diseases in whatever way he/she desires, allowing them to tailor the system to their personal preferences with regards to both potential diagnoses and interface. In addition to giving a more detailed account of the patient's condition than a single human clinician would be able to, it could potentially give legitimate treatment suggestions the clinician may not have even considered. A varied range of treatment suggestions sharing common symptoms may even indicate multiple health conditions within the patient.

However, while a computer-based decision system may certainly be helpful in clinical decision making, it is not a substitute for human interaction. Computerized systems, at this stage, are still incapable of accurately assessing the often complex symptoms that a patient typically experiences. These systems are also error prone and subject to problems such as poor reliability and misuse by inexperienced clinicians. Finally, they lack the sense of user friendliness that is mandatory for both a patient's assurance and a clinician's understanding of the situation.

Nevertheless, Clinical Decision Support Systems are a very promising option with regards to aiding physicians in diagnostic and treatment-related decisions. Because of their inherently progressive nature, they are expected to continue evolving and remedying the various challenges/obstacles plaguing them. Thus, in due time, they are expected to become an even more viable method of patient decision making than they are currently now.

Acknowledgments

This work was supported in part by National Science Foundation grant IIS-1231742 and the National Institutes of Health grant R21CA175974.

Bibliography

[1] L.T. Kohn, J. M. Corrigan, and M.S. Donaldson. *To Err Is Human: Building a Safer Health System*. National Academy Press, 2000.

[2] Amit X. Garg, Neill Adhikari, Heather McDonald, Rosas-Arellano M., P. J. Devereaux, Joseph Beyene, Justina Sam, and R. B. Haynes. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review. *Journal of the American Medical Association*, 293(10):1223–1238, 2005.

[3] Mary Moore and Kimberly A. Loper. An introduction to clinical decision support systems. *Journal of Electronic Resources in Medical Libraries*, 8(4):348–366, 2011.

[4] R. A. Miller, M. B. First, and L.J. Soffer. The internist-1/quick medical reference project-status report. *Western Journal of Medicine*, 145(6):816–822, 1986.

[5] E. S. Berner, G. D. Webster, A. A. Shugerman, J. R. Jackson, J. Algina, A. L. Baker, E. V. Ball, G. C. Cobbs, V. W. Dennis, E. P. Frenkel, L. D. Hudson, E. L. Mancall, C. E. Rackley, and O. D. Taunton. Performance of four computer-based diagnostic systems. *New England Journal of Medicine*, 330(25):1792–1796, 1994.

[6] Edward Shortliffe. *Computer-Based Medical Consultations: MYCIN*. Elsevier, 2012.

[7] Eta S. Berner. *Clinical Decision Support Systems: Theory and Practice*. Springer, 2007.

[8] Thomas A. O'Callaghan, James Popple, and Eric McCreath. Shyster-MYCIN: a hybrid legal expert system. *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 103–104, 2003.

[9] Rebecca S. Crowley and Olga Medvedeva. An intelligent tutoring system for visual classification problem solving. *Artificial Intelligence in Medicine*, 36(1):85–117, 2006.

[10] Michael J. Lincoln, C. W. Turner, P. J. Haug, H. R. Warner, J. W. Williomson, O. Bouhaddou, Sylvia G. Jessen, Dean Sorenson, Robert C. Cundick, and Morgan Grant. Iliad training enhances medical students' diagnostic skills. *Journal of Medical Systems*, 15(1):93–110, 1991.

[11] Homer R. Warner, Peter Haug, Omar Bouhaddou, Michael Lincoln, Homer Jr. Warner, Dean Sorenson, John W. Williamson, and Chinli Fan. Iliad as an expert consultant to teach differential diagnosis. *Proceedings/the... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, pages 371–376, 1988.

[12] Imad M. Tleyjeh, Hesham Nada, and Larry M. Baddour. VisualDX: Decision-support software for the diagnosis and management of dermatologic disorders. *Clinical Infectious Diseases*, 43(9):1177–1184, 2006.

[13] G. O. Barnett, J. J. Cimino, J. A. Hupp, and E. P. Hoffer. DXplain: An evolving diagnostic decision support system. *Journal of the American Medical Association*, 258(1):67–74, 1987.

[14] P. L. Elkin, M. Liebow, B. A. Bauer, S. Chaliki, J. Wahner-Roedler, D. Bundrick, Mark Lee, Steven Brown, David Froehling, Kent Bailey, Kathleen Famiglietti, Richard Kim, Ed Hoffer, Mitchell Feldman, and G. Barnet. The introduction of a diagnostic decision support system (DXplain) into the workflow of a teaching hospital service can decrease the cost of service for diagnostically challenging diagnostic related groups (DRGS). *International Journal of Medical Informatics*, 79(11):772–777, 2010.

[15] William F. Bond, L. M. Schwartz, K. R. Weaver, D. Levick, M. Giuliano, and M. L. Graber. Differential diagnosis generators: An evaluation of currently available computer programs. *Journal of General Internal Medicine*, 27(2):213–219, 2012.

[16] Mark L. Graber and Ashlei Mathew. Performance of a web-based clinical diagnosis support system for internists. *Journal of General Internal Medicine*, 23(1):37–40, 2008.

[17] P. Ramnarayan, A. Tomlinson, A. Rao, M. Coren, A. Winrow, and J. Britto. Isabel: A web-based differential diagnostic aid for paediatrics: Results from an initial performance evaluation. *Archives of Disease in Childhood*, 88(5):408–413, 2003.

[18] Guilan Kong, Dong-Ling Xu, and Jian-Bo Yang. Clinical decision support systems: A review on knowledge representation and inference under uncertainties. *International Journal Of Computational Intelligence Systems*, 1(2):159–167, 2008.

[19] B. Yegnanarayana. *Artificial Neural Networks*. PHI Learning Pvt. Ltd., 2009.

[20] J. M. Jerez-Aragones, J. A. Gomez-Ruiz, J. Munoz-Perez, and E. Alba-Conejo. A combined neural network and decision trees model for prognosis of breast cancer relapse. *Artificial Intelligence in Medicine*, 27(1):45–63, 2003.

[21] M. R. Brickley, J. P. Shepherd, and R. A. Armstrong. Neural networks: A new technique for development of decision support systems in dentistry. *Journal of Dentistry*, 26(4):305–309, 1998.

[22] David E Goldberg. *Genetic Algorithms*. Pearson Education India, 2006.

[23] Hyunjin Shin and Mia K. Markey. A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples. *Journal of Biomedical Informatics*, 39(2):227–248, 2006.

[24] Latha Parthiban and R. Subramanian. Intelligent heart disease prediction system using canfis and genetic algorithm. *International Journal of Biological, Biomedical and Medical Sciences*, 3(3):157–160, 2008.

[25] Randolph A. Miller. The anatomy of decision support during inpatient care provider order entry (CPOE): Empirical observations from a decade of CPOE experience at Vanderbilt. *Journal Of Biomedical Informatics*, 38(6):469–485, 2005.

[26] K.E. Bizovi, B. E. Beckley, M. C. McDade, A. L. Adams, R. A. Lowe, A. D. Zechnich, and J. R. Hedges. The effect of computer assisted prescription writing on emergency department prescription errors. *Academic Emergency Medicine*, 9(11):1168–1175, 2002.

[27] S. B. Flexner and J. Stein. *The Random House College Dictionary, Revised Edition*. Random House, Inc., 1988.

[28] R. A. Miller. Why the standard view is standard: People, not machines, understand peoples' problems. *Journal of Medicine and Philosophy*, 15(6):581–591, 1990.

[29] Mayo Clinic. Mayo clinic depression test. http://www.mayoclinic.org/diseases-conditions/depression/basics/tests-diagnosis/con-20032977 Accessed: 2014-08-23.

[30] Robert A. Greenes. *Clinical Decision Support: The Road Ahead*. Academic Press, 2006.

[31] Dean F. Sittig, Adam Wright, Jerome A. Osheroff, Blackford Middleton, Jonathan M. Teich, Joan S. Ash, Emily Campbell, and David W. Bates. Grand challenges in clinical decision support. *Journal of Biomedical Informatics*, 41(2):387–392, 2008.

[32] R. L. Engle Jr. Attempts to use computers as diagnostic aids in medical decision making: A thirty-year experience. *Perspectives in Biology and Medicine*, 35(2):207–219, 1991.

[33] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 2003.

[34] G. F. Luger and W.A. Stubblefield. *Artificial Intelligence and the Design of Expert Systems*. Benjamin/Cummings Publishing, 1989.

[35] Ida Sim. Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, 8(6):527–534, 2001.

[36] M. A. Minsky. A framework for representing knowledge. In *The Psychology of Computer Vision*, McGraw-Hill, 1975.

[37] F. Pinciroli, C. Combi, and G. Pozzi. Object oriented DBMS techniques for time oriented medical records. *Informatics for Health and Social Care*, 17(4):231–241, 1992.

[38] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[39] Y. Shahar and M.A. Musen. A temporal-abstraction system for patient monitoring. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 121, 1992.

[40] C.F. Aliferis, G.F Cooper, R.A. Miller, B.G Buchanan, R. Bankowitz, and N. A. Guise. A temporal analysis of qmr. *Journal of the American Medical Informatics Association*, 3(1):79–91, 1996.

[41] S. M. Weiss, C. A. Kulikowski, S. Amarel, and A. Safir. A model-based method for computer-aided medical decision making. *Artificial Intelligence*, 11(1):145–172, 1978.

[42] W. Long. Medical diagnosis using a probabilistic causal network. *Applied Artificial Intelligence: An International Journal*, 3(2-3):367–383, 1989.

[43] R. Patil, P. Szolovitz, and W. Schwartz. Causual understanding of patient illness in medical diagnosis. *Seventh International Joint Conference for Artificial Intelligence*, 81:893–899, 1981.

[44] H. A. Heathfield and J. Wyatt. Philosophies for the design and development of clinical decision support systems. *Methods of Information in Medicine*, 32:1–8, 1993.

[45] Louis G. Iasilli. Federal food, drug, and cosmetic act. *St. John's Law Review*, 13(2):425–437, 2014.

[46] V.M. Brannigan. Software quality regulation under the safe medical devices act of 1990: Hospitals are now the canaries in the software mine. *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, 1991:238–242.

[47] K. W. Goodman. *Ethics, Computing and Medicine: Informatics and the Transformation Of Health Care*. Cambridge University Press, 1997.