Clinical Decision Support System
Project 2 Report

Mohammed Saleem
Introduction
Ever since computer has emerged health professionals have dreamed of the day that computers will contribute in healthcare domain; assist them in so many ways, for instance storing all patient records for easy tracking, not only that but mainly helping them to make critical decisions or explore through a huge set of diseases and symptoms in considerable amount of time, in addition provide accurate and sufficient results in most cases. In Addition It has drawn to the attention that the number of medical errors caused by physicians has increased significantly; which is generally the result of a conclusion of the wrong diagnosis, a prescription of an incorrect treatment or the inefficient execution of the right treatment, not mentioning the huge gap between the demand for physicians to the actual number of available physicians in practice, therefore there is insufficient time for diagnose or treatment.
Clinical Decision Support System (CDSS) is one of the important filtering systems; this system will provide the physicians with the tools that will assist them in the diagnosis, picking the correct treatment or evaluate a therapy. This system is not intended to replace physicians rather it should be used by the physicians to improve health care quality. The tool provides qualitative and quantitative results from the input data regarding to a patient.

Literature
In this section we will explore the different CDS Systems that were developed from the early times that it has brought up to professional’s attention in 1950’s, to one of the successful systems that is nowadays by healthcare specialists, these systems range from specific medicine problem to generalized medicine, e.g. de Dombal's system for acute abdominal pain [de Dombal et al., 1972] was designed to diagnose diseases related to the abdominal body parts. And of course there are other systems developed for general medicine practice, these systems vary in terms of efficiency, accuracy and in performance. In addition we will explore some of the successful commercialized CDSS that are used until today in the healthcare domain.
One of the key point of the problem that these systems face, is the fact that on average each diseases have between 75 to 100 findings associated with it (symptoms, laboratory reports, etc), however these finding that are associated with a disease are not unique, in fact they are overlapping and nested, this is the bottleneck that confront all systems, many system has came up with algorithms to model this problem, for instance a tree that will connect finding along with disease in an way to avoid confusing and minimize as much as possible the output, in other words if the system is not determined which disease is causing this group of finding, a new set of tests are suggested or some questions are provided to the physician to help in understanding the disease better. In addition we realize the need for a dynamic system the will

a. De Dombal's system
developed at university of Leeds in the early 1970’s by deDombals and his associates. They studied the diagnoses process and developed a computer-based decision aids using Bayesian probability theory [Musen, 2001]. They have collected information for thousands patients, their system used sensitivity, specificity and disease-prevalence data for various signs, symptoms, and test results to calculate (using Bayesian probability theory) the probability of seven possible explanations for acute abdominal pain. To keep
the program manageable, the program made the assumptions of (1) conditional independence of the findings for the various diagnoses and (2) mutual exclusivity of the seven diagnose. They have tested the proposed system on a case study of 304 it showed that it has an accuracy of 91.8 percent meanwhile the clinicians were able to construct a 65 to 80 percent from the given case study. However the system wasn’t able to deliver the same accuracy level when it was tested in different domains.

b. INTERNIST-I

it was a broad-based computer-assisted diagnostic tool developed in the early 1970’s at the University of Pittsburgh as an educational experiment [Miller et al., 1982; Pople, 1982]. The system was designed to capture the expertise of just one man, Jack D. Myers, MD, chairman of internal medicine in the University of Pittsburgh School of Medicine. The INTERNIST-I project represented fifteen person-years of work, and by some reports covered 70-80% of all the possible diagnoses in internal medicine.

Data input into the system by operators included signs and symptoms, laboratory results, and other items of patient history. The system didn’t use Bayesian probability theory instead they used a powerful ranking algorithm to reach diagnoses in the domain of internal medicine. The heuristic rules that drove INTERNIST-I relied on a partitioning algorithm to create problems areas, and exclusion functions to eliminate diagnostic possibilities. These rules, produce a list of ranked diagnoses based on disease profiles existing in the system’s memory. When the system was unable to make a determination of diagnosis it asked questions or offered recommendations for further tests or observations to clear up the ambiguity. INTERNIST-I worked best when only a single disease was expressed in the patient, but handled complex cases where more than one disease was present poorly. This was because the system exclusively relied on hierarchical or taxonomic decision-tree logic which linked each disease profile to only one “parent” disease class. In (fig.1), a tree structure is shown of a knee replacement surgery, Probabilities has been assigned to each branch of each node of the tree, also the patients valuations of outcomes measured in years of perfect mobility are assigned to each branch of the tree.

![Fig.1 Decision tree for knee replacement surgery](image)
c. **MYCIN**

It was a rule-based expert system designed to diagnose and recommend treatment for certain blood infections (antimicrobial selection for patients with bacteremia or meningitis) [Shortliffe, 1976]. It was later extended to handle other infectious diseases. Clinical knowledge in MYCIN is represented as a set of IF-THEN rules with certainty factors attached to diagnoses (fig.2). It was a goal-directed system, using a basic backward chaining reasoning strategy resulting in exhaustive depth-first search of the rules base for relevant rules though with additional heuristic support to control the search for a proposed solution. MYCIN was developed in the mid-1970s by Ted Shortliffe and colleagues at Stanford University. It is probably the most famous early expert system, described by Mark Musen as being "the first convincing demonstration of the power of the rule-based approach in the development of robust clinical decision-support systems" [Musen, 1999].

![Fig.2 IF-THEN rules example used in MYCIN system](image)

---

d. **DXplain**

Designed by the Laboratory of Computer Science at the Massachusetts General Hospital, work on DXplain began in 1984 with a first version being released in 1986. Today, it contains a database with crude probabilities for over 4,900 clinical manifestations that are associated with over 2,200 unique diseases, yielding a total of over 230,000 unique finding-disease interconnections [LCS MGH Harvard Medical School].

It has the characteristics of both an electronic medical textbook and a medical reference system. In its reference or case analysis mode, DXplain accepts a set of clinical findings (signs, symptoms, laboratory data) to produce a ranked list of diagnoses which might explain (or be associated with) the clinical manifestations. It also provides justification for why each of these diseases might be considered, suggests what further clinical information would be useful to collect for each disease, and lists what clinical manifestations, if any, would be unusual or atypical for each of the specific diseases.

DXplain generates ranked differential diagnoses using a pseudo-probabilistic algorithm; each clinical finding entered into DXplain is assessed by determining the importance of the finding and how strongly the finding supports a given diagnosis for each disease in the knowledge base. Using this criterion, DXplain generates ranked differential diagnoses.
with the most likely diseases yielding the lowest rank. Using stored information regarding each disease’s prevalence and significance, the system differentiates between common and rare diseases.

e. **QMR - Quick Medical Reference**

Developed by University of Pittsburgh and First DataBank, Inc., San Bruno, CA based on INTERNIST-I, Quick Medical Reference (QMR) is an in-depth information resource that helps physicians to diagnose adult diseases. It provides electronic access to more than 750 diseases representing the vast majority of the disorders seen by internists in daily practice as well a compendium of less common diseases. QMR uses more than 5,000 clinical findings to describe the features of diseases in the QMR knowledge base [Lemaire, 1999]. Findings include medical history, symptoms, physical signs, and laboratory test results. Laboratory test results are subdivided into three categories based on increasing levels of cost and invasiveness. QMR findings represent abnormal conditions, e.g., "Abdomen Pain Severe" or "Blood Hepatitis B Virus by Polymerase Chain Reaction." Every disease profile included in the QMR knowledge base is the result of an extensive review of the primary medical literature. Consultation with experts is used to resolve any inconsistencies or deficiencies found in published reports. QMR is used in in-hospital and office practice. Windows versions of QMR were available for single and multiple users; a hand-held version was under development and may have been released.

**The Role of Computer in Decision Support**

The diagnostic process—deciding which questions to ask, tests to order, or procedures to perform, and determining the value of the results relative to associated risks or financial costs. Thus, diagnosis involves not only deciding what is true about a patient but also what data are needed to determine what is true. Even when the diagnosis is known, there often are challenging management decisions that test the physician’s knowledge and experience: Should I treat the patient or allow the process to resolve on its own? If treatment is indicated, what should it be? How should I use the patient’s response to therapy to guide me in determining whether an alternate approach should be tried or, in some cases, to question whether my initial diagnosis was incorrect after all? Biomedicine is also replete with decision tasks that do not involve specific patients or their diseases. Consider, for example, the biomedical scientists who are using laboratory data to help with the design of their next experiment, or the hospital administrator who uses management data to guide decisions about resource allocation in his hospital. Although we focus on systems to assist with clinical decisions, we emphasize that the concepts discussed generalized to many other problem areas as well.

A clinical decision-support system is any computer program designed to help health professionals make clinical decisions. In a sense, any computer system that deals with clinical data or medical knowledge is intended to provide decision support. It is accordingly useful to consider three types of decision-support functions, ranging from generalized to patient-specific.
a. Tools for information management
Health-care information systems and information-retrieval systems are tools that manage information. Specialized knowledge-management workstations are under development in research settings; these workstations provide sophisticated environments for storing and retrieving clinical knowledge, browsing through that knowledge much as we might page through a textbook, and augmenting it with personal notes and information that we may need later for clinical problem solving. Information-management tools provide the data and knowledge needed by the clinician, but they generally do not help her to apply that information to a particular decision task. Interpretation is left to the clinician, as is the decision about what information is needed to resolve the clinical problem.

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

c. Tools for providing patient-specific recommendations
Such programs provide custom-tailored assessments or advice based on sets of patient-specific data. They may follow simple logics (such as algorithms), may be based on decision theory and cost–benefit analysis, or may use numerical approaches only as an adjunct to symbolic problem solving. Some diagnostic assistants (such as DXplain [Barnett et al., 1987] or QMR [Miller et al., 1986]) suggest differential diagnoses or indicate additional information that would help to narrow the range of etiologic possibilities. Other systems suggest a single best explanation for a patient’s symptomatology. Other systems interpret and summarize the patient’s record over time in a manner sensitive to the clinical context [Shahar & Musen, 1996]. Still other systems provide therapy advice, rather than diagnostic assistance [Musen et al., 1996].

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

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

b. The Mode for Giving Advice

Like the abdominal pain program and MYCIN, most decision-support programs have assumed a passive role in giving advice to clinicians [Reggia & Turhim, 1985]. Under this model, the practitioner must recognize when advice would be useful, and then must make an explicit effort to access the computer program; the decision-support system waits for the user to come to it.

The clinician then describes a case by entering data, and requests a diagnostic or therapeutic assessment. There are also technologies, such as the HELP system, that play a more active role, providing decision support as a byproduct of monitoring or of data management activities; such systems do not wait for physicians or other health workers specifically to ask for assistance. A great appeal of such systems is their ability to give assistance to health-care workers without requiring laborious data entry by the clinicians themselves. Such capabilities are possible only because the system’s decision logic is integrated with a comprehensive database of patient information that is already being gathered from diverse sources within the health-care institution. Because practitioners generally do not request assistance from such systems, but instead receive it whenever monitored patient data warrant it, one challenge is to avoid generating excessive numbers of warnings for minor problems already likely to be understood. Otherwise, such “falsepositive” advisory reports can generate antagonistic responses from users and can blunt the usefulness of those warnings that have greater clinical significance.
c. Style of Communication
Decision-support systems have tended to operate under one of two styles of interaction: the consulting model or the critiquing model. In the consulting model, the program serves as an advisor, accepting patient-specific data, asking questions, and generating advice for the user about diagnosis or management. For example, MYCIN was an early example of a program that adopted the consulting approach. In the critiquing model, on the other hand, the clinician has a preconceived notion of what is happening with a patient or what management plan would be appropriate. The computer then acts as a sounding board for the user’s own ideas, expressing agreement or suggesting reasoned alternatives. A pioneering example of a critiquing system is ATTENDING, a stand-alone program that critiques a patient-specific plan for anesthetic selection, induction, and administration after that plan has been proposed by the anesthesiologist who will be managing the case [Miller, 1986]. Such critiquing systems meet many physicians’ desires to formulate plans on their own, but to have those plans double-checked occasionally before acting on them. In the critiquing style, the program focuses more directly on the plan in which the physician is interested. The critiquing model also can be applied in an active monitoring setting. For example, the HELP system monitors physicians’ drug-therapy decisions, and can suggest alternate approaches that may be preferable [Evans et al., 1986]. Similarly, the HyperCritic system [van der Lei & Musen,1991] can offer suggestions regarding how primary-care physicians might improve their management of patients with hypertension by performing a behind-the-scenes analysis of the patients’ computer-based record at the time of each clinic visit.

d. Underlying Decision-Making Process
a wide variety of techniques has been used in the design and implementation of decision-support systems. The simplest logics have involved problem-specific flowcharts designed by clinicians and then encoded for use by a computer. Although such algorithms have been useful for triage purposes and as a didactic technique used in journals and books where an overview for a problem’s management has been appropriate, they have been largely rejected by physicians as too simplistic for routine use [Grimm et al., 1975]. Because computers were traditionally viewed as numerical calculating machines, people had recognized by the 1960s that they could be used to compute the pertinent probabilities based on observations of patient-specific parameters (as long as each had a known statistical relationship to the possible disease etiologies). Large numbers of Bayesian diagnosis programs have been developed in the intervening years, many of which have been shown to be accurate in selecting among competing explanations of a patient’s disease state [Heckerman & Nathwani, 1992]. As we mentioned earlier, among the largest experiments have been those of deDombal and associates in England [de Dombal et al., 1972], who adopted a simple Bayesian model that assumed that there are no conditional dependencies among findings (for example, that the presence of a finding such as fever never affects the likelihood of the presence of a finding such as chills). More recent work on the use of belief networks for automated decision making has demonstrated that it is practical to develop more expressive Bayesian systems in which conditional dependencies can be modeled explicitly, rather than ignored.
Figure 3. Assessment, Decision Making, and Cost-Effectiveness Ladder
CDSS Components

Most of Clinical Decision Support System consists of four main interconnected components that interact with each other to achieve certain task, which are Inference Engine (IE), knowledge Base (KB), Explanation Module and Working Memory (fig 3.),

![CDSS Components Diagram](image)

The IE is the main part of any CDS System. The IE uses the knowledge on the system and the knowledge about the patient to draw conclusions regarding certain conditions. The IE controls what kind of actions need to be taken by the system. For example, it determines the route of alerts and reminders in an alerting system, or the conclusions to be displayed in a diagnostic system. The inference engine operates according to the algorithm which they were implemented in. The knowledge used by the IE is represented in the Knowledgebase. Knowledge bases may be built with the help of a domain expert or by an automated process. In the first case, a knowledge engineer (expert on building KBs) with the help of a clinical domain expert creates, edits, and maintains the KB. In an automated process, knowledge is acquired from external resources such as databases, books, and journal articles by a computer application or. Creating such knowledge bases can be a complex task. Fortunately, tools have been created to facilitate the acquisition and elicitation of KBs. An example of such a tool is Protégé [Musen MA, 1995,1998] a knowledgebase development environment. The collection of patient data may be
stored in a database or may exist in the form of a message. This collection is known as “working memory.” Patient data may include demographics (i.e., date of birth, gender), allergies, medications in use, previous dental or medical problems, and other information. The last component, the explanation module, is not present in all CDSS. This module is responsible for composing justifications for the conclusions drawn by the IE in applying the knowledge in the KB against patient data in the working memory.

CDSS can work on synchronous mode; that is, the application communicates directly with a user who is waiting for the output of the system. A typical example is a system that checks for drug-drug interactions or possible patient allergy to a medication when a provider is writing a prescription. In asynchronous mode, CDSS perform their reasoning independently of any users awaiting its output. An example is the generation of a reminder for an annual visit for checkup and hygiene. CDSS can be classified as open- or closed loop systems. In an open-loop system, the CDSS draws the conclusions but takes no action directly of its own. An application that generates an alert or reminder is an example of such systems. The final decision on the action to be taken, if any, is made by the clinician. In the closed-loop system, the action can be implemented directly without the need of a human. Other important types of CDSS are event monitors, consultation systems, and clinical guidelines. An event monitor is a software application that receives copies of all data available in electronic format in an institution and uses its knowledge base to send alerts and reminders to clinicians when deemed appropriate [Hripcsak G,1996]. In consultation systems, a clinician enters details of a case (e.g., patient demographic information, clinical history, physical examination findings, and test results) into the system, and the system, in turn, provides a list of problems that may explain the case and suggest actions to be taken.

**Conclusion**

Clinical decision support system has been discussed and developed since the 1950’s where the first paper about clinical support system was published, although it hasn’t been a big success in its earliest times; it is today considered one of the hug successes of AI. The emergence of CDSS as paradigm has contributed significantly in the healthcare domain through various applications and systems. You cannot step into health institute without noticing the dramatic effect that these machines caused to improve health care, from the computers used by professionals to pull out the patient electronic records in seconds to the small device that fits in the physician’s pocket which help them in the process of diagnoses. It was with out a doubt that inevitably these systems will be used in all health care institutes. These systems come with a great range of specification, with different domain applications, some are specific and most are derived towards general practice.
References:


Musen MA. Stanford Medical Informatics: uncommon research, common goals. MD Comput. 1999 Jan-Feb;16(1):47-8, 50.


Miller RA, Masarie FE, Myers JD. Quick Medical Reference (QMR) for diagnostic assistance. MD Comput. 1986;3:34-38.


