Clinical Decision Support Systems in Biomedical Informatics and their Limitations

Alberto De la Rosa Algarín
University of Connecticut, Storrs
alberto.delarosa.algarin@engr.uconn.edu

April 2011

Abstract

Clinical decision support systems can be categorized in three types: information management systems, focusing attention systems and patient specific recommendation systems. Characteristics like data validation, workflow integration and performance are necessary for the success of a CDSS. Several limitations exist with current systems that delay a wider adoption. These limitations range from lack of a seamless integration to knowledge sharing. Advancements on standards and novel input methods try to overcome these obstacles.

1 Introduction

Clinical decisions can be categorized in two types: diagnosis and diagnostic process. Diagnosis decisions are done analyzing data to determine the cause of sickness. Diagnostic process, or management, decides which questions to ask [1].

Three requirements must be met in order to perform excellent decision-making. First, accurate data. Inaccessible, bad data is useless for decision making. Good data is of no use if there is no previous knowledge of how to apply it properly. Second, pertinent knowledge. Overload of data hinders the decision making process. In several health care settings, constant monitoring of a patient is necessary. This produces a large amount of information, from which maybe a subset is only necessary. Third, appropriate problem-solving skills. In order to connect the previous two requirements together, good problem-solving skills are needed. The main goal with clinical decision support systems is to emulate a clinician's thought process during decision making, yielding more accurate results.

Clinical decision support systems (CDSS) can be clustered in three different types [1].

- *Information Management Systems*, for storing and retrieving clinical knowledge. The interpretation of such knowledge is left to the clinician.
- *Focusing Attention Systems*, which alert the user of possible conflicts or problems that might have been missed.
- *Patient Specific Recommendation Systems*, which provide personal

assessment of a patient, usually following simple logic rules.

In this paper we first explore the origin of CDSS. In section 2 we take a look at the requirements CDSS must satisfy. In section 3 we observe the existing systems and how they attempt to satisfy these requirements. In section 4 we discuss the shortcomings of current systems. In section 5 we propose several ideas that could improve current systems.

2 Clinical Decision Support Systems

In this section we define what a decision support system is. Then we see the first systems created and then we look at the different types of CDSS.

2.1 Definition

A decision support system is defined as *a system in which one or more computers and computer programs assist in decision-making by providing information* [2]. A CDSS is more specifically defined as software, a program created to help clinicians reach decisions with better accuracy.

2.2 History

The possibility of CDSS appeared in [3], proposing that the medical diagnosis involves processes that can be systematically analyzed. With the use of three mathematical disciplines (symbolic logic, probability and value theory), the foundations of medical diagnosis can be understood.

Symbolic logic emphasizes the importance of considering the combination of symptoms with the combination of diseases. Probability concepts are inherent in medical diagnosis because health care professionals can never give a 100% accurate diagnosis. Value theory arises because clinicians must offer the treatment that will improve health the most while still inside the bounds of social constraints.

The first prototypes are shown in [4]. Issues with logistics, scientific shortcomings and a lack of integration to the workflow prevented the widespread adoption of these early prototypes.

Several CDSS improved on the previous shortcomings, managing to do breakthrough discoveries in the use of Bayesian reasoning and differential diagnosis. The most significant, early CDSS were *Leeds Abdominal Pain System*, *MYCIN* and *HELP*.

MYCIN [5], a consultation system designed for appropriate management of patients with infections, used production rules to represent the knowledge of infectious diseases. *Leeds Abdominal Pain System* [6], designed to help reach more accurate diagnoses, used sensitivity, specificity and disease-prevalence data from various symptoms. Combined with Bayesian probability theory, the probability of seven possible explanations for acute abdominal pain were shown. *HELP* [7] has the ability to alert clinicians when abnormalities in the patient record are noted. With the use of the Arden syntax and medical logic modules, HELP checks wether new patient data matches the preset criteria.

2.3 CDSS Types

CDSS can be categorized in three types: information management, focusing attention and providing patient-specific recommendations.

Information management tools are designed to provide environments for storing and retrieving medical knowledge [1]. They can also offer intuitive ways of traversing through that information, while incrementing the quality of it with notes and second degree data that might be needed for future decisions. In these types of systems, the decision is left to the clinician.

Tools for focusing attention offer ways to alert clinicians know when some unforeseen conflict might arise, such as potentially dangerous drug interactions [8, 9]. These systems follow simple logic to reach their intended behavior.

Patient-specific recommendation systems are designed to offer advice to a single patient using that patient's previous medical history. While these systems might also follow simple logic, they can also use decision theory, cost-benefit analysis and rough set theory [10].

3 Requirements of a CDSS

CDSS must satisfy a set of requirements so their acceptance in the health care domain increases. These requirements include unresolved questions in both science and logistics, and range from patient data (acquisition and validation of), medical knowledge (including its modeling, elicitation, representation and reasoning), system performance to the integration in the clinician's workflow [1]. In this section we will look at the expectations of a CDSS in terms of those requirements.

3.1 Patient Data

Patient data is the source of information clinicians use, combined with their medical knowledge, to decide between possible diagnoses. CDSS must be able to acquire and validate data in a way that is seamless to the clinician and secure in terms of the patient's privacy.

3.1.1 Acquisition

There is no standard way to acquire patient data. Current methods range from keyboard and speech to human intermediaries between the clinicians and computers. While technologies like the keyboard and natural speech are proven to work, they often do not fit in the workflow of the health care specialist. Keyboards disrupt the process completely, and such data entry is often left to intermediaries like medical assistants or secretaries. Natural language processing has the potential to extract clinical information [11], but the dilemma of allowing unrestricted data input affects the user interface and the structure and encoding of the data, which are obstacles not easy to overcome.

Some more recent and novel methods have been created [12], but have yet to be proven useful in the traditional care workflow. Thus a good CDSS should be able to capture data without disrupting the workflow. Data acquisition would need to have a combination of speech, graphics and concurrent data monitoring that does not obstruct the clinician's routine.

3.1.2 Validation

Despite all the existing standards for medical terminologies, there is no coding system able to capture all the details of care given by clinicians [13]. There is also no coding system able to capture the subtle differences in a patient's illness and medical history [13].

Existing electronic health records offer the capability of storing most of the patient's information, but not all the relevant data. While still useful, the current solutions should be viewed as incomplete solutions. To reach a suitable form of patient data validation, there would need to be a level of consensus between the different coding standards. Some attempts have been made to reach that consensus [14], but still need to improve as they do not incorporate all the existing standards.

CDSS should be dynamic enough to work with highly and poorly detailed data, while at the same time providing reliable recommendations.

3.2 Knowledge

In order to reach medical decisions, there needs to be a combination of two types of medical knowledge: low-level and high-level knowledge [15]. Low-level knowledge represents the structure and function of the body, diseases and their causes, treatments. This type of knowledge is acquired during the academic career of the clinician. High-level knowledge is generally gathered from clinical experience and allows the clinicians to make accurate decisions.

Medical knowledge needs to be modeled, elicited, represented and reasoned efficiently in order to make a CDSS worthy of adoption.

3.2.1 Modeling

Modeling of medical knowledge is not a trivial task. Deciding what patient data is relevant, identifying concepts and relationships, and utilizing a strategy for solving require a great amount of modeling.

Some methodologies, like Common KADS [16], have been adopted in a broad level. Frameworks like these aim to allow decision support system builders to model knowledge, regardless of the underlying decision making methodology implemented.

Others, like CASNET [17], are used in designing consultation programs for the diagnosis of long-term treatment diseases. In the case of CASNET, a set of general decision-making strategies is used in conjuction to a class of causal-associational models. This allows the model representation and decision-making procedures to be generalizable to other medical domains.

3.2.2 Elicitation

A good CDSS should be able to evoke useful knowledge seamlessly. This implies methods that facilitate the development and maintenance of knowledge-bases. Current programs that obtain knowledge work directly with the clinical expert, avoiding the need of a middleman.

In order to create domain-specific knowledge evocation, developers need to first create their model of the intended application area for the target decision-support system and then either program that model by hand into the tool or enter the model into a meta-tool. Protege is such a tool [18, 19].

3.2.3 Representation

Health care specialists use mental models of relationships between body parts and organs when they interpret data or plan medical care. Trends and progression of sickness or medicine results are other types of medical knowledge difficult to represent in computer systems. The interpretation of these types of knowledge is intuitive to humans.

The ideal CDSS should be able to store factual or inferential knowledge, and at the same time it should be able to emulate human intuition in terms of data interpretation.

3.2.4 Reasoning

Health care experts have the ability of knowing what knowledge is useful and how to properly apply that knowledge. Computer systems have the potential to store great amounts of factual knowledge. CDSS should be able to discern what knowledge presented is useful and it should know how to apply it.

3.3 System performance

The amount of medical knowledge is increasing rapidly. It is expected from computer systems and software to maintain a "gold standard" for performance. Thus CDSS must be able to provide results in an instant while using the most recent patient data and medical knowledge available. The system must also give accurate results every time its used. The effects of suggesting a wrong diagnosis could be fatal.

3.4 Integration to workflow

Workflow integration can make or break the acceptance of a CDSS. Hospitals and clinics use multiple computers optimized for different tasks, and the challenges of integration are tied to issues of networking and user interfaces.

Data input, data presentation and system performance are enough to determine how well the system integrates in the workflow. CDSS must satisfy these previous requirements in order to be seamlessly integrated in the workflow.

4 Modern CDSS

There exist several CDSS, with uses ranging from image-based software to systems that help the diagnosis of cardiovascular diseases. These CDSS trade one of the requirements for others in order to maintain a level of performance good enough to be useful. In this section we describe several of the modern CDSS (late 1980's - present), how they work, their current uses and their limitations.

4.1 Pathfinder

Pathfinder is an expert system that assists surgical pathologists with the diagnosis of lymph-node diseases [20]. The system uses probability and decision theory to acquire, represent, manipulate and explain uncertain medical knowledge.

It uses the same method as a pathologist during diagnosis. First, it identifies and quantifies features. Second, it constructs a differential diagnosis. Third, it decides what additional features to evaluate and what costly tests employ to narrow the differential diagnosis. It is important to notice that the system does not recommend diagnosis since such actions are sensitive to the utility model implemented [20].

The user interface in Pathfinder is menu based and mouse driven. The initial screen has multiple windows: *feature category*, *observed features* and *differential diagnosis*. The *feature category*, window, which displays the categories of features that are known to the system.

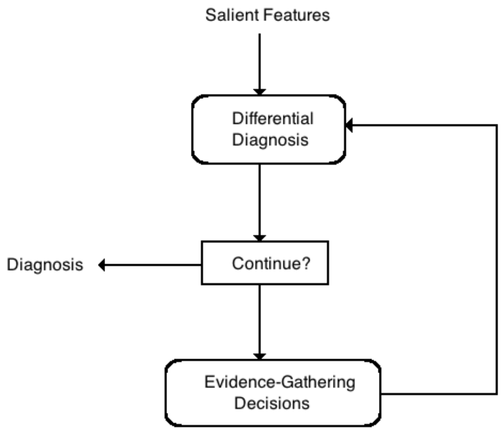


Figure 1: *Pathfinder's deductive reasoning model*

The *observed features* displays feature-instance pairs that the pathologist will observe. The *differential diagnosis* window displays the list of possible diseases and their probabilities. The user is able to select a feature category (by clicking the mouse), which prompts Pathfinder to display a list of features for that category. The same gesture is used to enter particular features.

The second version of Pathfinder (Pathfinder II) uses belief networks instead of the model used in the first version. Pathfinder II increased the expected utility of patients who received a diagnosis by an average of $6,000 per case (in contrast to Pathfinder I) [21]. This means that the development of Pathfinder II would pay for itself if it were use at least in three cases.

4.2 ILIAD

The ILIAD consultant utilizes a number of inferencing mechanisms to emulate the strategy of a medical expert in working up with a patient. The knowledge in ILIAD is represented in Bayesian and Boolean frames. These frames permit the use of sensitivities and specificities to describe the relationship of a disease to its manifestations and provide a basis for explaining its conclusions [22]. ILIAD has four basic components: the inference engine, the user interface, the data driver and the best information algorithm.

The inference engine is independent of subject area. The engine controls the communication with the user, is responsible of evoking the needed knowledge frames, requests information needed and explain conclusions to the user.

The user interface allows the user to control the operation of the application to meet the required needs. It achieves this with the use of drop down menus. It also adds the possibility of adding data, which allows the user to type one or more words or partial words describing a medical finding.

The data driver creates a pointer to each frame added to the knowledge-base. This pointer is created from each dictionary item used by that frame to the appropriate slot in the frame. This allows the system to automatically evoke the logic that uses each piece of information.

The best information algorithm used by ILIAD uses a scoring mechanism and is described in [22].

ILIAD's current use is as a teaching tool for medical students, where particular cases are simulated so that students learn how to diagnose.

4.3 DiagnosisPro

DiagnosisPro is a tool that uses differential diagnosis. It reminds the user of diagnostic possibilities in an effort to minimize medical errors. Its user interface allows the input of one or more findings or conditions, then the system generates a hierarchical list of diagnosis from its knowledge-base [23].

The knowledge-base includes over 11,000 diseases, 30,000 findings and 300,000 relationships. The information is taken from known medical resources such as Harrison's Principles of Internal Medicine, Oxford Textbook of Medicine, JAMA and others.

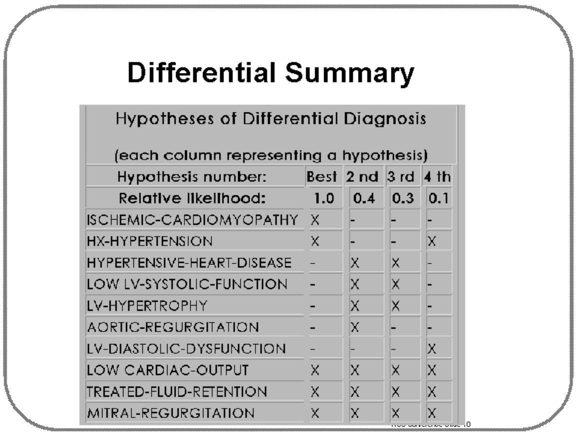
The system is designed to help clinicians reach a more accurate diagnosis with the use of recent developments and advancements of diseases and medical history. It reminds practitioners of all the possibilities that might have been forgotten.

4.4 HDP

HDP (Heart Disease Program) is a computer system that assists the physician in the task of differential diagnosis and anticipating the effects of therapy in the domain of cardiovascular disorders. It uses a knowledge-base combined with a physiologic model, hemodynamic function and dysfunction and probabilities and constraints. Its user interface is menu driven. HDP also uses a differential diagnoses generator.

The knowledge-base has around 200 physiologic state nodes, covers the common hemodynamic problems and covers diseases influencing hemodymanics [24]. The input interface uses history, vital signs, physical exam results, laboratory findings and hemodynamic data. The differential diagnosis engine uses a diagnostic mechanism that combines a Bayesian network with the constraints presented by severities of the states and the temporal relations of causality. The algorithm of differential diagnosis is described in [24].

This algorithm provides a performance increase over existing models, but the method is still heuristic. To predict the effects of therapy, HDP has a mechanism that uses equations for the hemodynamic relationships and a signal flow technique to calculate the likely quantitative steady-state change for all parameters given changes in therapies. This mechanism effectively captures the hemodynamic effects of the therapies on which it has been tested for a variety of pathophysiologic conditions.



Differential Summary

Hypotheses of Differential Diagnosis

(each column representing a hypothesis)

Hypothesis number:	Best	2 nd	3 rd	4 th
Relative likelihood:	1.0	0.4	0.3	0.1
ISCHEMIC-CARDIOMYOPATHY	X	-	-	-
HX-HYPERTENSION	X	-	-	X
HYPERTENSIVE-HEART-DISEASE	-	X	X	-
LOW LV-SYSTOLIC-FUNCTION	-	X	X	-
LV-HYPERTROPHY	-	X	X	-
AORTIC-REGURGITATION	-	X	-	-
LV-DIASTOLIC-DYSFUNCTION	-	-	-	X
LOW CARDIAC-OUTPUT	X	X	X	X
TREATED-FLUID-RETENTION	X	X	X	X
MITRAL-REGURGITATION	X	X	X	X

Figure 2: *HDP: Example of a Differential Summary*

4.5 CKS

The NHS CKS (Clinical Knowledge Summaries) is a service that provides ready access to digestible clinical knowledge. It aims to help clinicians make evidence based decisions about patient's healthcare, while at the same time providing strategies of how to use these decisions. It builds on the existing PRODIGY knowledge-base. Its uses a web-based user interface, making it possible for users to access from anywhere around the world.

CKS provides knowledge on clinical topics about common acute and chronic diseases and their prevention [25]. It offers quick answers, summaries on how to manage common clinical scenarios. It also offers detailed answers which link recommendations to the evidence on which they are based.

The methodology that the system uses is as follows. It finds and considers the relevant literature on the topic (guidelines, reviews, studies and policy documents) from the HPA and the DoH. It then summarizes the knowledge and makes it available in one location [25].

4.6 DXplain

Developed by the Laboratory of Computer Science at the Massachusetts GEneral Hospital, it combines characteristics of an electronic medical textbook with characteristics of a medical reference system. DXplain has a case analysis mode, where the clinician inputs several findings like symptoms and laboratory test results, to produce a list of probable diagnoses. It provides a rank for each of the possible diagnosis and an explanation on why that diagnosis should be considered [26].

DXplain can provide information of different diseases, emphasizing signs and symptoms. It also provides the etiology, the pathology and the prognosis of each of the different diseases. The knowledge-base for DXplain includes over 2,400 diseases and over 5,000 symptoms, signs, laboratory data and other clinical findings [27]. It works by assigning two numbers to each disease-finding pair, which function as a description for the relationship (one is the frequency in which the finding occurs in the disease and the other is the degree to which the presence of the findings suggest consideration of the disease).

4.7 VisualDx

VisualDx is a JAVA-based decision support tool developed as a point-of-care reference. Just like an atlas with color photographs, VisualDx is used to reference a visual presentation and to confirm diagnosis. One of the main functions is the facilitation of image matching for the end user by combining graphical search tools, a computerized knowledge-base of relationships between findings and diagnoses, and thousands of digital images [28].

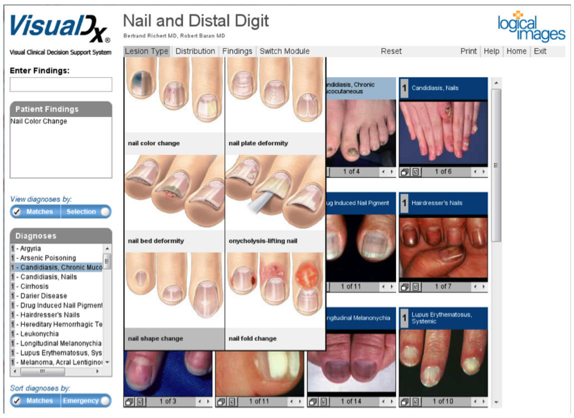


Figure 3: *Example of VisualDx's User Interface*

It is used in clinical care to develop differential diagnoses based upon morphologic and patient driven search [28]. Clinicians can enter patient descriptors and lesion morphologies, resulting in rapid assistance with differential diagnosis. VisualDx also increases clinician awareness of, knowledge about, and skills in the recognition of chemical warfare, bioterrorism, and radiation injuries.

VisualDx focus is on infectious diseases. It consists of several modules that are relevant to infectious diseases specialists and public health epidemiologists.

4.8 INTERNIST-1/QMR Project

INTERNIST-1 and QMR (Quick Medical Reference, its successor) are designed to provide diagnostic assistance in general internal medicine. Both rely on the INTERNIST-1 knowledge-base, which comprehensively describes 572 diagnoses in internal medicine, and the system recognizes more than 4,000 possible patient findings. The knowledge-base also includes more than 4,000 links detailing the casual, temporal and probable interrelationships among the 572 disorders [29]. While QMR is the successor of INTERNIST-1, both programs function differently.

QMR acts as an information tool, providing users with multiple ways of reviewing and manipulating the diagnostic information in the program's knowledge-base. At the lowest level, QMR can be viewed as an electronic medicine book that can assist users in creating hypotheses in complex patient cases. INTERNIST-1 functions only as a high-powered diagnostic consultant program.

QMR provides several key features that try to tackle some of the existing problems of CDSS adoption. First, the QMR completer. The user interface tries to reduce the amount of typing necessary and the amount of errors due to data entry thanks to a "completer" program. This allows users to enter the names of various patient findings and diagnoses already known. The completer then provides a list of possible hits.

Second, the system functions as a low-level information retrieval tool. It achieves this using two options: display a disease profile's findings and links, or display the differential diagnosis of a single finding. This last option gives the user the possibility to display a list of possible diagnoses associated with any of the findings in the knowledge-base.

Third, QMR can be used as an intermediate-level information management tool. An option can be used to connect a seemingly unrelated finding to a disease in a specified organ system.

Last, there is the option to use the system as a high-level information management utility. The case-analysis mode allows a user to enter up to 95 positive and 95 negative findings from a case [29]. These findings are used to create hypothesis of different types, with the intent of allowing the clinician to assert which of the diagnoses is present in the patient.

4.9 EON System

EON consists of four general purpose software components. The first of those components interprets abstract protocol specifications to construct appropriate patient-specific treatment plans. The second component infers from time-stampted data some higher-level, interval-based, abstract components. The third component performs time-oriented queries on a time-oriented patient database. Finally the last component allows acquisition and maintenance of protocol knowledge in a manner that facilitates efficient processing both by users and computers [30].

The design principles that create a base for the EON system are problem-solving methods and domain ontologies. Because of the issues with unpredictable behavior of rule-based representations, and the difficulties of long term maintenance of knowledge-bases, the group responsi-

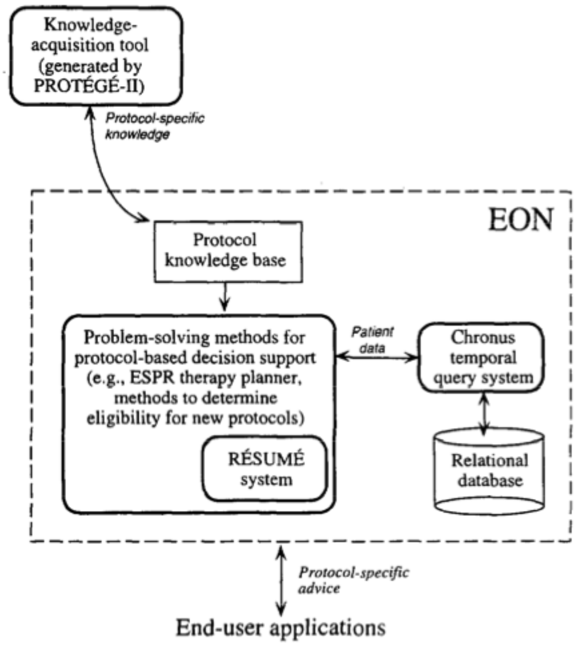


Figure 4: *EON System architecture*

ble for EON developed PROTEGE-II, a framework for building knowledge-based systems [30]. In PROTEGE-II, the data requirements of problem-solving methods are satisfied by the domain-specific knowledge that instantiates the domain ontology for a particular application.

5 Limitations

Existing clinical decision support systems suffer from limitations that are difficult to overcome. Obstacles ranging usability to security hinder the adoption rate of these systems. In this section we discuss the existing shortcomings of different areas.

5.1 Patient Role

The patient's role in current support systems is a "passive" one. The patient is seen as the source of information from which the system feeds. The user of the systems is the health care specialist, and no system makes a real attempt to change the nature of that role.

The lack of a well-defined role for the patient generates several question that vary from case to case. First, *should the patient's privacy affect his/her right to know?*. Second, *is there a real need for a patient to access information offered by decision support systems?*. Third, *should some of the capabilities of decision support systems be built in PHR's (Personal Health Records)?*.

Such questions do not only have implications in a moral or ethical sense, but can also provide evidence for legal cases. As an example, take the hypothetical scenario of a patient that goes to a hospital for care. After giving his/her medical history, the patient is assigned a doctor. The doctor utilizes a CDSS to help diagnose the patient, but at the very end discards the recommendations of the system and assigns a diagnosis that he/she feels is correct. With this diagnosis, medications are prescribed to the patient. In a few weeks, the patient is back because the medications did not cure his/her illness. It turns out the CDSS recommended the correct diagnosis from the very start. If the patient were to know that the CDSS made the correct diagnosis, a lawsuit could start because of malpractice.

We also have to notice that social constraints make health care professionals practice medicine tailored to each patient's economic state. Following the previous example, if a CDSS designed to recommend the most effective-cheapest treatment is used, the patient would want to know all the recommendations that system has to make.

In the end we have a dilemma. The patient will surely want to know every bit of detail the clinician knows, but clinicians would want to withhold information to the patient because the effects of releasing it could be disastrous.

5.2 Usability

Usability is the biggest hurdle current CDSS have to overcome. Modern systems do very little to integrate in the workflow seamlessly. This limitation is the result of smaller shortcomings in areas of human-computer interaction, user-interface and input methods, and performance of the support systems themselves.

First, *should user-interfaces offer the full detail of diagnosis or only the pertinent information and hide everything else?* This question would have different answers depending on the clinician responding. A busy health care professional would answer that only the pertinent information is needed. Less busy professionals would naturally appreciate a high level of detail. [31] argue that if a guideline cannot be fitted in a single screen, clinicians will not be happy about using it. At the same time, additional information should only be displayed when the clinician needs it.

Second, *how should input be handled? Is a keyboard enough? How about natural speech?* Doctors do not like to modify the usual workflow to input data. Because of this, input methods have changed over time. Methods like [12] aim to capture speech, writing and typing at the same time, in an effort to bridge the gap between non-digital and digital data acquisition.

Accuracy performance limits the usability of these systems. Studies have been done to determine how well CDSS as QMR and ILIAD perform in emergency scenarios. Results show that systems like these have the same success of correct diagnosis in emergency settings as in any other clinical settings [32]. Accuracy still is not high enough to make these systems useful as arbiters of individual cases.

[31] found that the speed of the system is an attribute users hold in high regard. Studies found that the primary factor for user satisfaction is the speed, rated higher than quality improvement aspects. Users perceived physician order entry primarily as an efficient technology. If the decision support takes too long to appear, it will be useless.

5.3 Knowledge Sharing and Maintenance

Knowledge-bases are specific to each CDSS. One of the boasting points of current solutions is the amount of diagnoses their product's knowledge-base has. Naturally we ask, *should the knowledge be shared?*

Having a centralized knowledge-base, or at least a framework to share knowledge, would improve accuracy and reliability of recommendations. The limitations that make such a scenario impossible do not lie in the logistical infrastructure, but in the coding standards.

Several coding standards exist, but there is no standard of standards. With such a situation, it is left to each system's developers to implement support for the remaining standards.

Thus the obstacle to overcome in order to achieve knowledge sharing is to reach a compromise between standards. This way the burden of development is greatly reduced, and interoperability between systems would be easier to implement.

Maintaining the knowledge within the system and managing the individual pieces of the system are critical to successful delivery of decision support [31]. The amount of effort needed to achieve this is considerable, and current attempts at solving this rest in the requirement of individuals updating the knowledge base periodically.

5.4 Security

The state of security in existing CDSS is achieved by only allowing access to the primary clinician. Systems are designed to provide a certain level of recommendation, and whoever has access to the system is able to obtain the same recommendation.

Should nurses not see the same recommendations surgeons see? At the same time, *what would happen if the patient's role transforms from a passive to an active one? Should the patient be omniscient?*

In contemporary health care, other professionals (nurses, pharmacists, radiologists, surgeons) are an equal part of the patients well-being as the physician. So it is logic to think that CDSS's security should be role based.

There is ongoing research on different security methodologies for electronic health records, policy-based and technology-based solutions [33]. Technology-based solutions aim to use special hardware or software to limit access. This can be achieved with the use of smart cards, special identifiers and the likes. Policy-based solutions are more focused on model legislation that define right of access, data element definition, procedures for access and release, and monetary fines for abuse.

It would be ideal if any of these two approaches could be designed and implemented in existing and future CDSS.

6 Conclusion

We have seen the origin of clinical decision support systems. We have also learned about the requirements of a CDSS. While looking into existing solutions, we found several limitations that impede the widespread adoption of the systems and limitations that have a profound scientific and legal base.

The patient's role in CDSS is not defined clearly. The patient, while blindingly trusting health care professionals with personal information, is only the source of information for decision support systems. If the patient's role were to be an active one, this would have scientific and personal implications. The patient has the right to know everything about the care given, but not earlier than the clinician. At the same time, this would have legal implications.

Widespread adoption of CDSS is hindered by the lack of usability. Hurdles in user-interface, input methods and performance need to be cleared in order to start a widespread adoption. Performance could be improved with knowledge sharing. The high number of standards in knowledge coding do not permit developers of CDSS to implement each and every standard.

Last, security in CDSS needs an improvement. Several approaches have been made on electronic health records, approaches that could be imported to CDSS in an effort to augment the current status of security.

7 Future Work

A long road lies ahead of CDSS. There needs to be advancements in the integration to the workflow, performance, knowledge sharing and security in order to advance the adoption of CDSS.

In order to further integrate CDSS into the workflow, several ideas arise. The use of current tablets like the Apple iPad could bridge the difference between non-digital and digital data acquisition. The iPad could replace current paper charts, and with the use of a capacitive pen and OCR technology, data acquisition would be natural.

Performance and knowledge sharing are tied limitations. With the creation of a standard of standards, all current coded knowledge would fall into a single common umbrella. This will permit current and future CDSS to be implemented with support to that only standards, trickling down the compatibility with previous ones.

Existing alternatives exist in terms of security. The implementation of either of the current approaches suggested for electronic health records would greatly improve CDSS's security.

References

[1] E. H. Shortliffe, *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. Springer, 3 ed., 2006.

[2] Dictionary.com, "Collins english dictionary - complete unabridged 10th edition," Mar 20011.

[3] R. Ledley and L. Lusted, "Reasoning foundations of medical diagnosis," *Science*, 1959.

[4] H. Warner, A. Toronto, and L. Veasy, "Experience with baye's theorem for computer diagnosis of congenital heart disease," *Annals of the New York Academy of Sciences*, vol. 115, p. 558, 1964.

[5] E. Shortliffe, "Computer-based medical consultations: Mycin," 1976.

[6] F. De Dombal, D. Leaper, J. Staniland, A. McCann, and J. Horrocks, "Computer-aided diagnosis of acute abdominal pain," *British Medical Journal*, vol. 2, no. 5804, p. 9, 1972.

[7] H. Warner, *Computer–Assisted Medical Decision-Making*. Academic Press, Inc. Orlando, FL, USA, 1979.

[8] R. Evans, R. Larsen, J. Burke, R. Gardner, F. Meier, J. Jacobson, M. Conti, J. Jacobson, and R. Hulse, "Computer surveillance of hospital-acquired infections and antibiotic use," *Jama*, vol. 256, no. 8, p. 1007, 1986.

[9] D. Tatro, R. Briggs, R. Chavez-Pardo, L. Feinberg, J. Hannigan, T. Moore, and S. Cohen, "Online drug interaction surveillance," *American Journal of Health-System Pharmacy*, vol. 32, no. 4, p. 417, 1975.

[10] Z. Pawlak, "Rough set approach to knowledge-based decision support," *European journal of operational research*, vol. 99, no. 1, pp. 48–57, 1997.

[11] G. Hripcsak, C. Friedman, P. Alderson, W. DuMouchel, S. Johnson, and P. Clayton, "Unlocking clinical data from narrative reports: a study of natural language processing," *Annals of Internal Medicine*, vol. 122, no. 9, p. 681, 1995.

[12] B. Ahier, "The pen is mightier than the keyboard."

[13] J. Cimino, "Review paper: coding systems in health care," *Methods Inf Med*, vol. 35, no. 4-5, pp. 273–84, 1996.

[14] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. Database Issue, p. D267, 2004.

[15] J. Wyatt, "Use and sources of medical knowledge.," *Lancet*, vol. 338, no. 8779, p. 1368, 1991.

[16] R. De Hoog, R. Martil, B. Wielinga, R. Taylor, C. Bright, W. Van De Velde, C. Georges, and C. Gustavsson, "The Common KADS model set," 1994.

[17] S. Weiss, C. Kulikowski, and A. Safir, "A model-based consultation system for the long-term management of glaucoma," in *International Joint Conference on Artificial Intelligence*, pp. 826–832.

[18] N. Noy, R. Fergerson, and M. Musen, "The knowledge model of Protege-2000: Combining interoperability and flexibility," *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 69–82, 2000.

[19] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, and S. Tu, "The evolution of Protégé: an environment for knowledge-based systems development," *International Journal of Human-Computer Studies*, vol. 58, no. 1, pp. 89–123, 2003.

[20] D. Heckerman, E. Horvitz, and B. Nathwani, "Toward normative expert systems: Part I. The Pathfinder project," *Methods of information in medicine*, vol. 31, no. 2, pp. 90–105, 1992.

[21] D. Heckerman and B. Nathwani, "Toward normative expert systems: Part II. Probability-based representations for efficient knowledge acquisition and inference," *Methods of Information in medicine*, vol. 31, no. 2, pp. 106–116, 1992.

[22] H. R. Warner, P. Haug, O. Bouhaddou, M. Lincoln, H. W. Jr, J. W. Williamson, and C. Fan, "Iliad as an expert consultant to teach differential diagnosis," 1988.

[23] DiagnosisPro.com.

[24] W. Long, "Medical diagnosis using a probabilistic causal network," *Applied Artificial Intelligence*, 1989.

[25] NHS, *Clinical Knowledge Summaries: Overview*.

[26] G. O. Barnett, K. T. Famiglietti, R. J. Kim, E. P. Hoffer, and M. J. Feldman, "Dxplain on the internet," 1998.

[27] M. S. Packer, J. J. Cimino, E. P. Hoffer, G. O. Barnett, R. Kim, and S. Zatz, "Updating the dxplain database," *1988*.

[28] I. M. Tleyjeh, H. Nada, and L. M. Baddour, "Visualdx: Decision-support software for the diagnosis and management of dermatologic disorders," *Clinical Infectious Diseases*, vol. 43, September 2006.

[29] R. A. Miller, M. A. McNeil, S. M. Challinor, F. E. Masarie, and J. D.

Myers, "The internist-1/quick medical reference project - status report," December 1986.

[30] M. A. Musen, S. W. Tu, and A. K. D. Y. Shahar, "Eon: A component-based approach to automation of protocol-directed therapy," *Journal of the American Medical Informatics Association*, vol. 3, pp. 367–388, December 1996.

[31] D. Bates, G. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton, "Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality," *Journal of the American Medical Informatics Association*, vol. 10, no. 6, pp. 523–530, 2003.

[32] M. A. Graber and D. VanScoy, "How well does decision support software perform in the emergency department?," 2003.

[33] D. M. Grzybowski, "Patient privacy: The right to know versus the need to access," vol. 26, pp. 53–54, September 2005.