Extending Contemporary Decision Support System Designs to Patient-Oriented Systems

George C. Scott, Leslie A. Lenert, M.D., M.S.
Division of Clinical Pharmacology, Stanford University School of Medicine, Stanford, California

Decision support systems for patients can benefit from adopting knowledge engineering-based architectures. In this paper, we describe how decision support systems for patients differ from decision support systems for health professionals and knowledge engineering principles that can be used to improve the efficiency of developing patient support systems. We discuss a five-step process model for patient-computer dialogue and its incorporation into an architecture based on knowledge engineering ontologies. The architecture's components are grouped into transient and persistent application layers that support a general framework for patient decision support. The implementation of the object-based model using a relational database management system is also discussed.

INTRODUCTION AND BACKGROUND

Knowledge engineering systems have traditionally focused on the application of domain ontologies and knowledge bases for guideline-based physician support. However, little has been done to address process of the generalization of decision support system for patients. Recent advances in support systems designed for health professionals can be applied to greatly improve the efficiency of developing automated, intelligent patient-oriented systems.

Both types of systems require the ability to maintain and utilize domain ontologies and knowledge bases. Both seek to optimize an outcome based upon a given context by using a well defined set of algorithms and heuristics. Moreover, both need to be able to adequately and to intelligently explain their results to the user in a meaningful manner.

Support systems for health professionals such as ONCOCIN[1,2] and Protégé-II[3,4], which are based on guidelines, are usually intended to determine the best protocol for the patient given their assignment to a guideline. Patient support systems such as SecondOpinion[5] intend to produce insight into the best therapy option for a patient given their preferences for different outcomes. In this sense, both types of support systems attempt to determine what plan of action will result in the best outcome. However, while past systems typically provided guidance based upon what is best for the average patient, patient-oriented systems must focus on advice that is custom tailored for the individual.

Like the support systems for health professionals, the patient support systems must use specific tools or instruments to acquire information from the user. Systems like Protégé-II use knowledge acquisition tools to derive the ontology and knowledge base for systems. These are generally used by domain experts to develop the domain ontology.

Decision support systems for patients based on normative frameworks have a goal of generating insight into the decision process[6], rather than providing a critique of a decision[7,8] or finding the optimal treatment based on a treatment protocol. These systems achieve this goal by educating the patient about the options and outcomes for a disease, assessing their preferences for the outcomes with psychometric methods (standard gamble, visual-analog scale) and revealing to the patient the implications of his or her preferences for their choice of a treatment. Throughout this process, the system attempts to help the patient to understand the significant outcomes, how his or her preferences may differ from other's and the effect those preferences should have on the specific treatment option being evaluated by the system[6].

Although both types of systems share many common design needs, they do have specific differences. Unlike a support system that often needs objective information from a health professional, normative decision support systems for patients must also be able to acquire and manage subjective value measurements from patients. The goal of said management is to ensure accuracy and reduce the burden of questioning[6]. Once the ontology for a decision support system for health professionals is established, the system usually assumes that the knowledge acquired is accurate and complete. However, normative patient support systems must expect and be able to deal with errors and inconsistencies in preference measurements which can occur during the elicitation process[9]. Since normative patient systems are frequently driven by calculation from the underlying decision model, to maintain such systems, developers needs only to manage a small set

1091—8280/98/$5.00

of methodologies and ontologies specific to decision analysis.

SecondOpinion is a prototype system for interactive WWW-based normative decision support system for patients, which was described previously[5]. In this paper, we describe a general set of extensions of this framework based on the knowledge-engineering architecture of Protégé-II. This set of extensions is design to facilitate rapid-development of WWW-based decision support system for patients driven by a validated decision model. We discuss the framework architecture and the implementation of its object-based hierarchies in a relational database model.

FRAMEWORK ARCHITECTURE

In order to generalize decision support applications we propose the use of formal ontologies in the conceptual model. We began with domain and method ontologies characterized in the description of Protégé-II. To these we added two additional hierarchies: the interface ontology and the decision support ontology. In addition, the domain ontology was divided into a disease (knowledge) and a patient (information) ontology.

Within our application, ontologies are grouped into two conceptual layers: transient and persistent. The content of each of the layers is based upon a discussion semantic. A discussion is modeled as being a process composed of two parts: the underlying ideas, information and knowledge upon which the dialogue itself is based and the expression and acquisition of that knowledge. During a discussion, information is constantly translated between these two representations. Each of the layers is enumerated by an ontological set that describes the relationships and contents of each of the components. This architecture and the relationships between the frameworks are illustrated in Figure 1.

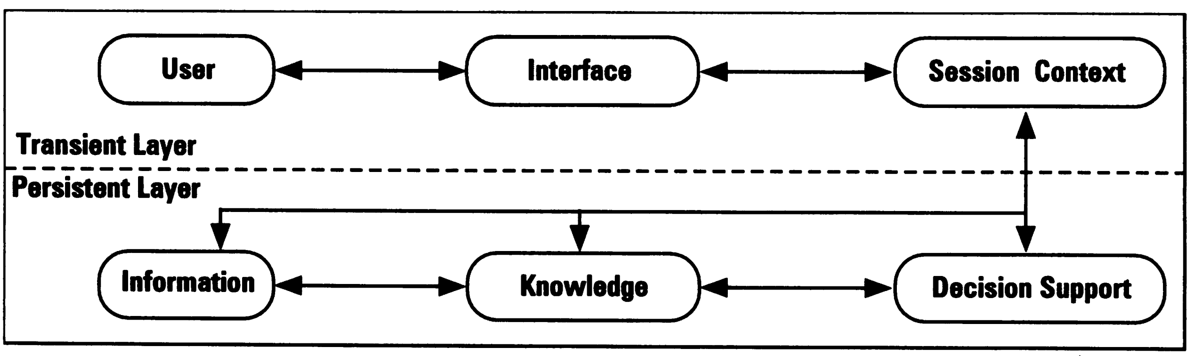
Transient Layer

The transient layer consists of the components necessary for the expression and acquisition of the domain information. Just as a discussion is a single possible representation of the transfer of a given set of knowledge and information, an instance of the transient layer represents one of many possible representations of interactions with the system. Information in the transient layer is stored and accessed in a semantic fashion. That is, the information is managed in a language-based sense. It has the ability to model a discussion and provide feedback.

Interface Engine. The interface engine provides the ability to translate the current user's session into a viewable, intelligent interaction with the user. It can generate a dynamic interface based on the information and knowledge stored in the system.

HTML is currently being used as the interface modeling language since it is based on a text stream and can be easily modified at run-time. This permits template files, analogous to HTML stylesheets, to be easily created and modified to allow customization of content. Because HTML is a structured language, it is possible to easily represent it in an object-oriented paradigm. By managing the interface as a separate context, it is possible to use a different interface standard or multiple standards at the same time without having to re-implement the remainder of the system. This could, for example, allow such things as the ability to have the same system drive both an HTML and a Java interface.

Session Manager. The session manager provides a framework structure to the user interaction and allows the system to apply an intelligent model to the computer dialogue. It does this by managing a finite state representation of a cyclic discussion model process. The decision process model consists of a hierarchy of five states that we previously described:



inconsistency resolution, knowledge reflection, education, knowledge acquisition and explanation[5]. The inconsistency resolution step allows for the correction of logical or semantic errors during user information acquisition. Unlike decision support systems for health professionals that may assume that the information entered by the user is accurate, a patient-oriented system must check to ensure that the measurements of preferences performed by the system yield data that are internally. Inconsistent responses that cannot be corrected imply that the system cannot learn about the preferences of the patient used the methods it has available and should refer the patient elsewhere for assistance. The knowledge reflection step provides the ability to periodically review the current state of the discussion and to explore the impact of the most recently updated information on the final recommendation. This allows for the outcome's sensitivity to a given piece of information to be reviewed after the information is acquired. Although this is not a formal sensitivity analysis, it can allow the user to follow the path taken by the program in reaching the final recommendation.

The education step allows the system to provide the user with information about the current topic in the discussion. It is important to make sure that the patient understands what aspect of the decision the system is exploring before it can ask the patient questions about it.

The knowledge acquisition step is when the system actively inquires, either directly or via an instrument, about information from the patient. The explanation step is the final and possibly most important part of the process. This is when the interface engine translates the explanation logic generated by the inference engine into a response.

Persistent Layer. The persistent layer consists of the actual knowledge and information that is necessary to conduct the interactive dialogue. It stores the information that is necessary to support any one of the possible representations of a discussion. Unlike the transient layer, information is stored and accessed in the persistent layer on a relational basis.

Inference Base. The inference base is responsible for the inferential computations and explanation logic generation that is necessary for decision support. It applies knowledge of the problem to a given instance of the problem. It can take domain knowledge, consider a specific set of constraints, and optimize. It then generates an explanation schema for its actions based on heuristics and assumptions used in the modeling and subsequent optimizations. Although similar to Protégé-II in approach, it is based on empirical and not symbolic algebra. The inference base is enumerated by the inference ontology below.

Information Base. The information base is the persistence of the information gathered through user interaction with the system. This information becomes the instantiation (parameters and constraints) of a given problem. Information about patients would also be stored in the information base. The schema of the database is determined by the patient ontology, which is described below.

Knowledge Base. The knowledge base is the structured persistence of the domain knowledge set necessary to make an "informed and knowledgeable" decision. This is the information needed to describe and explain the problem and its solution. Information regarding diseases that the system could consider and the issues related to therapy decisions for those diseases are stored in the knowledge base. The schema of the knowledge base is specified by the disease ontology described below.

Transient Layer Ontologies

The interface ontologiy and session context are used to describe and navigate, respectfully, the interface model for interactive decision support. It represents the parts of an interface in an object hierarchy. The structured nature of the HTML language greatly simplifies the application of the ontology to the creation and maintenance of an HTML interface.

Interface Ontology. The primary members of the interface ontology correspond with the HTML interface's components. The interface ontology hierarchy is used to describe the structure of an HTML interface and is used by the interface engine to build the dynamic interface.

Session Context. The session context is the specification of the states and their values used by the session manager to navigate the computer dialogue model with the patient and the underlying normative and assessment methods which direct said dialogue.

Persistent Layer Ontologies

The persistent layer ontology is used to describe the domain knowledge, instantiation parameters, and methodologies necessary for decision support. The separation of the domain ontology into a patient and disease ontology was done to emphasize the difference between the information and knowledge. The disease ontology represents a set of knowledge that is constant for all patients, whereas the patient ontology represents a set of information that is valid for only a given patient.

Patient Ontology. The patient ontology is used by the information base to model patient users of the system. It enumerates the information that each patient requires for a decision support session, including preference assessments, physician prognoses and previous recommendations.

Disease Ontology. The disease ontology is used by the knowledge base to model the diseases being considered by the system. It specifically enumerates the therapy options and expected health states for each disease. It also maintains different population contexts for a given disease. This allows for a single disease to have multiple sets of options and outcomes depending on the population being considered. A patient can be in only population at a time, but may change populations over time.

Decision Support Ontology. The decision support ontology is analogous to the method ontology of Protégé-II. It stores a set of analytic methods for simulation and modeling of a disease process to determine expected outcomes across a series of health states. There is also a set of methods that can be used to weight the outcome vector with the patient's preferences to obtain a quality adjusted joint outcome distribution.

IMPLEMENTATION

The system has been implemented in a relational database management system (RDBMS). Each object in the ontology hierarchies is used as a definition for a table in a RDBMS. The instantiations of the objects in the system ontologies are implemented as records in a table. This way, the properties of an object become the fields of the object's table. By using an object definition as the table definition, it is possible to use a RDBMS to store "arrays" of instantiated objects. Fields and relations are then used to store parent child relationships between the objects.

The main reason for this approach is that many legacy systems are based on RDBMS engines that might make integration with an object-based or object-oriented system difficult. By using simple tables, it is possible to represent the object hierarchy along with the system data and use the RDBMS engine already in place. Also, by leveraging existing enterprise-level technologies, it should be possible to easily scale-up such a support system for use by large groups of patients.

Knowledge acquisition tools could be used to automatically generate the table schemas necessary for the system. Thus, a user would be able to develop the ontologies necessary for the system using a knowledge acquisition tool and then generate the necessary tables in a RDBMS.

A drawback of using a relational model for the object-based hierarchy is the lack of an explicit inheritance methodology like that found in object-oriented languages. Since the object-based model currently only uses association and aggregation, but not inheritance, it is not a significant drawback at this time. It is possible, however, to emulate inheritance in a relational model.

DISCUSSION

The principles we outline in this paper focus upon both the user interaction as well as the underlying knowledge structures. The components of the system are divided into two application layers. The transient layer provides a representation of a discussion dialogue; the persistent layer provides a representation of the underlying knowledge, information and algorithms used in the discussion. A separation of the domain ontology into a disease and patient ontology better emphasizes the difference between domain specific versus instance specific information in the system. Finally, the addition of interface and dialogue ontologies provides the system with an intelligent model for user interaction. These tools are necessary for general models of decision support as a computer-patient dialogue.

Although there are differences between decision support systems for health professionals and decision support system for patients, patient support systems can benefit from advances in physician support system design. Decision support system for patients can also benefit from knowledge-engineering principles. By implementing the system in a relational database model, we believe that it will be possible to integrate new decision support applications with legacy information systems. By representing the ontological hierarchies in relational tables, it should be possible to transparently merge such system with existing systems.

There have been attempts in the past at developing general patient support systems, most notably the CHESS system[10,11]. The CHESS system was designed to function as a "shell" system to provide information services, communication services and decision support to patients and their peers in many different domains. Although found to be very effective and built with a well-developed functionality set, a formal ontology for such a system was never described. Unlike the CHESS system, which is a collection of patient-driven modules for exchanging and viewing information, we propose an integrated knowledge-based system that is model-driven and interactive. While the patient-driven model requires that the patient seek out information without knowing it value, a model-driven system is capable of providing the most "valuable" information first and allowing the patient to explore further. This should provide the patient with a more informative and user-friendly experience.

Since the patient decision support process is similar regardless of domain, it should be possible to construct a decision support system that is capable of providing interactive, model-driven decision support for patients, while allowing for the domain-specific knowledge to be represented at the time of system construction.

Evaluation of a knowledge-based intelligent decision support system for patients as a whole is difficult. The main factors in the benchmarking of such a system should be the usability and acceptance of the interface and the accuracy and adequacy of the domain knowledge. Although, these can be assessed via user questionnaires and expert review, respectfully, these are subjective values and can vary among groups.

The schemas presented are designed to form a foundation from which more detailed and use specific systems can be built. They are not intended to be a complete solution to every possible application, but are meant to be extensible. Since each of the system's components is designed to be self-contained, it should be possible to make changes without affecting the remainder of the system. This should make evaluation and maintenance of the different components much easier.

We plan to assess the robustness of the system through development and clinical assessment of support systems in several (including the BPH system SecondOpinion currently maintains). We also plan to test its ability to integrate with information management systems and determine its acceptance to the public.

ACKNOWLEDGEMENTS

This work was supported by the RK Richards Medical Student Research Assistant Fund, Department of Clinical Pharmacology, Stanford University and Grant LM05626-05 from the National Library of Medicine.

REFERENCES

1 Hickam DH, Shortliffe EH, Bischoff MB, Scott AC, Jacobs CD. The treatment advice of a computer-based cancer chemotherapy protocol advisor. Ann Intern Med 1985;103(6 (Pt 1)):928-36.

2 Kent DL, Shortliffe EH, Carlson RW, Bischoff MB, Jacobs CD. Improvements in data collection through physician use of a computer- based chemotherapy treatment consultant. J Clin Oncol 1985;3(10):1409-17.

3 Musen MA, Eriksson H, Gennari JH, Tu SW, Puerta AR. PROTEGE-II: a suite of tools for development of intelligent systems from reusable components. Proc Annu Symp Comput Appl Med Care 1994:1065.

4 Musen MA, Gennari JH, Eriksson H, Tu SW, Puerta AR. PROTEGE-II: computer support for development of intelligent systems from libraries of components. Medinfo 1995;8 Pt 1:766-70.

5 Scott GC, Cher DJ, Lenert LA. SecondOpinion: interactive Web-based access to a decision model. Proc AMIA Annu Fall Symp 1997:769-73.

6 Cher DJ, Lenert LA. Rapid approximation of confidence intervals for Markov process decision models: applications in decision support systems. J Am Med Inform Assoc 1997;4(4):301-12.

7 Miller PL, Blumenfrucht SJ, Black HR. An expert system which critiques patient workup: modeling conflicting expertise. Comput Biomed Res 1984;17(6):554-69.

8 Miller PL. Medical plan-analysis by computer. Comput Programs Biomed 1984;18(1-2):15-9.

9 Lenert LA, Morss S, Goldstein MK, Bergen MR, Faustman WO, Garber AM. Measurement of the validity of utility elicitations performed by computerized interview. Med Care 1997;35(9):915-20.

10 Gustafson DH, Bosworth K, Hawkins RP, Boberg EW, Bricker E. CHESS: a computer-based system for providing information, referrals, decision support and social support to people facing medical and other health-related crises. Proc Annu Symp Comput Appl Med Care 1992:161-5.

11 McTavish FM, Gustafson DH, Owens BH, Wise M, Taylor JO, Apantaku FM, et al. CHESS: An interactive computer system for women with breast cancer piloted with an under-served population. Proc Annu Symp Comput Appl Med Care 1994:599-603.