Automated Identification of Relevant Patient Information in a Physician's Workstation

Henri J. Suermondt, Ph.D.,[1] Paul C. Tang, M.D.,[1,2] Philip C. Strong, M.D.,[1]
Charles Y. Young, Ph.D.,[1] and Jurgen Annevelink, Ph.D.[1]

[1]Hewlett-Packard Laboratories, Palo Alto, California

[2]Stanford University School of Medicine, Stanford, California

The introduction of computer-based patient records offers an opportunity to improve the ability of physicians to browse the medical record and to monitor patient events. We describe a methodology to identify relevant information in the patient record. This methodology combines a patient-specific physiological model with functions to determine relevance of patient information. The model consists of a qualitative representation of physiological parameters and influences, custom-tailored to a particular patient's problems, medications, and test results. We describe two applications of this model in the context of an integrated physician's workstation: automatic linking of relevant patient information for configuration of user-interface displays, and monitoring of patient events to prevent oversight of noteworthy information.

INTRODUCTION

There has been increasing interest in the computer-based patient record (CPR), especially given the recent Institute of Medicine (IOM) study advocating the widespread adoption of the CPR in anticipation of future benefits [1]. These benefits include better quality of care, improved efficiency of care, and better data capture for use in resource-utilization review [2].

To maximize its potential benefits, the CPR must become a fully integrated part of the workflow of care providers, rather than simply an electronic version of the paper-based system. Health-care providers, particularly physicians, should be enabled to use computers for many aspects of their everyday practice, and have immediate access to relevant patient information at the time when they make, justify, and document their decisions. This need for physician-oriented systems provided the motivating drive for the Physician's Workstation project.

The Physician's Workstation (PWS) project is a long-term research effort to create and evaluate a workstation for use by physicians in primary care [3]. The project aims to provide physicians with integrated access to clinical information from diverse sources within the care environment. In addition, the goals of the project include the creation of integrated computer applications that help physicians improve their workflow and efficiency, the investigation of various forms of decision support in the setting of primary care, and the improvement of clinical information capture.

An important drawback of paper-based systems is the poor accessibility of information in the chart, even when the chart is available [1]. As a result, important information may be overlooked or ignored because it is not found easily in the record. The CPR provides us with the opportunity to provide safeguards to ensure that important information is not overlooked, and to facilitate the linkage of relevant information. In this paper, we discuss our approach to providing such functionality in the context of PWS.

Evaluation of Information in the Patient Context

Having information available on-line in a CPR is not very helpful if the physician who is using the system does not know how to retrieve the information. One can imagine an important laboratory-test result being "buried" in a maze-like record that can be traversed only by difficult menu navigation. In such a system, we may encounter problems of information access that are similar to the problems that plague the paper system: Even though the information is in the record, it is overlooked at the time when it should be considered for patient-care decisions.

As part of the decision-support functionality of PWS, we have designed a program called Radarserver that interprets patient-specific information as soon as it becomes available to the PWS system. Radarserver maintains a patient-specific physiological model, which is used to assess the relevance of incoming information given the patient context. Radarserver's model consists of a qualitative representation of physiological parameters and influences, custom-tailored to the patient's problems, medications, and test results. We describe the model in more detail in the following section.

The functionality that we derive from this patient-

0195-4210/92/$5.00 © 1994 AMIA, Inc.

specific model consists of (1) dynamic *linking* of relevant items in the patient record, and (2) *monitoring* of patient events to prevent oversights. The first function, linking relevant patient information, is intended to facilitate the navigation through the patient record, and to provide the user interface with intelligent defaults for the display of information. For example, when a physician chooses to review a particular problem, the system—within the limits of its model scope—identifies relevant risk factors, treatments, and test results. The second function, monitoring incoming information, serves as a safeguard to ensure that certain information is not overlooked by the user. The system monitors laboratory-test results for abnormalities and trends, and evaluates medication orders interactively for contraindications given the additional information in the patient record. We describe these functions in more detail in the section on use of the model.

THE PATIENT-SPECIFIC MODEL

The PWS physiological model consists of a directed graph in which the nodes represent physiological parameters and the arcs denote qualitative influences among parameters. The arcs are intended to capture causal and associational mechanisms by which perturbations to one parameter can affect others. This notion of a qualitative physiological model was derived from earlier work by Long [4], and was first adapted to the PWS context by Stanton and Tang [5]. Currently, the model consists of over 150 parameters and over 200 arcs, and spans elements of cardiovascular, pulmonary, renal, hepatic, hematologic, gastrointestinal, and endocrine physiology. Like other models that aim for a wide scope and use in clinical settings, the PWS model is continuously being refined and expanded.

In addition to the physiological parameters, we have represented a set of problems, medications, and tests. These entities are linked to parameters in the physiological model through *bindings*. Bindings reflect the influence of an entity on the qualitative state of a parameter relative to its normal state (e.g., *high*), as well as a qualitative notion of strength and probability. The bindings from items in the patient's record to the model are used to instantiate the generic physiological model for a particular patient's context. Currently, we have represented over 400 problems, 500 medications, and 150 laboratory tests. Our long-term goal is to have a comprehensive representation of the physiologically significant entities that are typically encountered for patients in a General Internal Medicine clinic.

Knowledge Acquisition

We have developed a set of tools to facilitate knowledge acquisition. These tools include graphical editors for the physiological model and for the problems, treatments, and tests, as well as a tool that can be used to obtain a detailed view of the influences on each parameter and the interactions among influences for a particular patient. To add an entity to the model, we combine information obtained from the medical literature with the clinical judgment of the two internists on our team (PCT and PCS). We first model the bindings to physiological parameters that are known to be affected by the entity, after which we test interactions with previously modeled entities.

Inference

To reason about a particular patient, Radarserver creates a version of the physiological model that is custom-tailored to the patient's context. We refer to this patient-specific version of the model as the patient's *model environment*. The model environment consists of a set of local environment descriptions of each parameter in the model, which describe the entities bound to the parameter as well as the qualitative value, strength, and likelihood of the bindings.

The object-oriented database underlying PWS is used to integrate locally entered patient information with information from a variety of external systems [6]. The first time a patient's model environment is created, the PWS object database server provides Radarserver with a batch of information that describes the state of the patient up to that time. Subsequently, the model environment is updated incrementally as patient events occur. Patient information can be updated either directly by PWS operations (for example, when a user orders a medication), or by additions to the database that occur asynchronously (for example, when a laboratory result is reported on an external computer system).

Whenever a user performs an update to the patient record using PWS, Radarserver monitors the interprocess communication between the user interface and the database [6, 7], and extracts any information that can be used in the patient's instantiation of the physiological model. In the case of updates to the patient record that take place externally, the PWS object database server continuously probes the externally maintained portions of the patient record, and notifies Radarserver whenever a change is detected.

When a patient event occurs, the program instantiates the event in the patient's model environment. First, Radarserver identifies the model entity (i.e., problem, medication, or laboratory test) that corresponds to the event. In our current prototype, we simply match the event name (e.g., the name of the medication that was ordered) against the known entity names of each type of patient event; we maintain a number of aliases for each entity. In the future, we envision making use of a vocabulary server—based on emerging standards—to ameliorate the problem of matching the user's terminology to our internal representation. After the proper entity has been identified, the entity's bindings to the physiological model are instantiated, and the effects of these bindings are propagated through the model. This propagation consists of reevaluating the arcs and bindings for which the qualitative strength is conditional on the state of the patient context, and updating the local environment descriptions of each parameter in the model.

After the patient's model environment has been updated to take into account the new event, Radarserver examines the model environment to determine whether the new element of the patient context affects the clinical relevance of any information. Relevance is determined by probing the environment of each parameter and determining whether there are any interactions among bindings—direct or propagated—that render entities relevant to one another, or that make an entity's effect on a parameter worrisome. We discuss the functionality resulting from this evaluation in more detail in the following section.

USE OF THE MODEL

Radarserver's functionality consists of two parts: (1) to respond to queries regarding clinical relevance, and (2) to monitor patient events, generating notifications when noteworthy developments are detected. We discuss these two tasks in turn.

Determining Relevance in Response to Queries

One of PWS' functions is to provide physicians with a temporal view of patient events. Thus, the user can view—in a single graphical display—the entire course of a clinical problem, including relevant medications and test results. Although it is possible for the user to configure such a display manually (by selecting from a set of lists the patient's problems, medications, and test results), Radarserver enables PWS to configure the display automatically, after the user has selected a particular entity of interest.

If the user selects a particular patient problem to view, the user interface generates a query for Radarserver to determine what is *relevant* to the problem for the patient. Radarserver probes the patient-specific instantiation of the physiological model, and uses it to identify the patient's *treatments* (current and past) that are likely to be of interest given the problem, as well as *laboratory tests* that can be helpful to diagnose and assess the problem and monitor responses to and side effects of its treatments. This information about relevant parameters is sent back to the user-interface server, which uses it to configure the display and show the information that is likely to be useful to the user.

If the user selects a medication, Radarserver identifies the patient's *problems* that could be affected by the medication (both positively and negatively), as well as the patient's available *laboratory-test results* that may be relevant in order to check the level of the medication, its clinical effectiveness, or its potential side effects.

Monitoring Patient Events

A second task of Radarserver is to "spot" the physician by monitoring patient events. When the program detects a noteworthy event, it generates a notification containing relevant information that might otherwise be overlooked during clinical decision making.

Rather than being triggered by a query, the monitoring task is initiated by the introduction of a new patient event (such as a medication order or renewal, or a laboratory result). When a change to the patient record is detected, the patient event is instantiated in the physiological model, as described earlier in this paper. After the changes to the patient's record have been reflected in the model environment, Radarserver determines whether there are any "alarming" situations demanding the attention of the patient's medical team. Examples of such situations are:

- The patient's known problems or laboratory results contraindicate the medication that is currently being ordered
- A laboratory result indicates an exacerbation of one of the patient's problems
- The patient's medications may be producing noteworthy adverse effects

In response to such a situation, Radarserver can notify one or more users with a message that summarizes a potential problem. The message mentions the patient event that triggered it, as well as a brief summary of other items in the patient record that are particularly relevant given the detected problem. This message is assigned a priority, depending on Radarserver's assessment of the seriousness and urgency of the situation.

Subsequently, the program, in conjunction with the PWS database, determines who needs to receive the message. If the patient's physician is currently using the system, the message is dispatched immediately, and appears in a special part of the PWS display. If none of the patient's physicians are using PWS—as may often be the case when laboratory results become available—the system can send the message by electronic mail, or, if the priority of the problem warrants it, can broadcast the message to the physician's pager.

When the receiver of such a notification message decides to view it in PWS, the program uses the query capability of Radarserver in order to immediately configure a patient-specific display that contains all the information that is relevant to the message. Thus, the original text message is complemented with a graphical overview of the relevant clinical information.

DISCUSSION

The appropriateness of Radarserver's responses is determined, to some extent, not by whether something was "truly" relevant or noteworthy, but rather, by the degree of user satisfaction (or annoyance) that resulted from the messages and relevance determinations. The program may occasionally generate a "false positive" (i.e., an item is shown even though the user does not find it helpful in monitoring a problem or treatment of interest) or a "false negative" (i.e., Radarserver did not report a particular item as relevant even though the user would like to have seen it). We intend to evaluate the appropriateness of the system's behavior—and the effects on users—as part of a rigorous clinical evaluation of PWS during the coming year.

With a poorly designed information system, the problems in retrieving clinical information at the time when patient-care decisions are made may persist even if the necessary data are available somewhere on the system. We developed Radarserver to help physicians identify and monitor relevant information without forcing them to pursue exhaustive menu traversals through an electronic record. Thus, Radarserver's model-based techniques, in conjunction with PWS' object database server and physician applications, are intended to help transform the CPR from simply an electronic form of the paper record, to an active document consisting of patient data as well as tools to browse, manage, and respond to patient information.

Acknowledgments

William M. Stanton has made important contributions to earlier versions of the PWS knowledge base and inference methods. We also thank Philippe De Smedt, Danielle Fafchamps, Curtis Kolovson, and Thuan Pham for their contributions to the PWS project.

Reference

[1] Institute of Medicine. The Computer-Based Patient Record: An Essential Technology for Health Care. National Academy Press, Washington, DC, 1991.

[2] G.O. Barnett. The application of computer-based medical-record systems in ambulatory practice. N Engl J Med **310** (1984) pp. 1643–1650.

[3] P.C. Tang, J. Annevelink, D. Fafchamps, W.M. Stanton, and C.Y. Young. Physician's workstations: Integrated information management for clinicians. In: P. Clayton, ed., Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, New York, 1991, pp. 569–573.

[4] W.J. Long, S. Naimi, M.G. Criscitiello, and S. Kurzrok. Reasoning about therapy from a physiological model. In: R. Salamon, A. Blum, and M. Jorgensen, eds., Proceedings of MEDINFO 86. Elsevier, Amsterdam, 1986, pp. 756–760.

[5] W.M. Stanton, and P.C. Tang. Knowledge-based support for a physician's workstation. In: P. Clayton, ed., Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, New York, 1991, pp. 649–653.

[6] J. Annevelink, C.Y. Young, and P.C. Tang. Heterogeneous database integration in a physician workstation. In: P. Clayton, ed., Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, New York, 1991, pp. 368–372.

[7] C.Y. Young, P.C. Tang, and J. Annevelink. An open systems architecture for development of a physician's workstation. In: Clayton, P., ed., Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care. McGraw-Hill, New York, 1991, pp. 491–495.