

An ICU Protocol Development and Management System

Jiabin Xie, Adwait Nerlikar,
John R. Glover
*Dept. of Electrical and Computer
Engineering, University of Houston*
glover@uh.edu

Bruce A. McKinley
*Dept. of Anesthesiology
Univ. of Texas - Houston Medical
School*
bmckinle@anes1.med.uth.tmc.edu

Abstract

Patient care is commonly managed by clinicians using general guidelines or specific protocols. Acute, severe illness may require care in an intensive care unit (ICU), with many aspects of care managed concurrently. Variation of any aspect of care among patients, clinicians, ICUs and institutions is well known, and is due to a lack of standardization in clinical decision making and inability to implement detailed, patient responsive protocols. Automation of protocols for specific aspects of care using computing technology and medical informatics/knowledge engineering principles offers a consistent, systematic way to implement a protocol by prompting bedside clinicians in a timely manner, monitoring a patient's responses to therapeutic interventions, and monitoring the decision logic used. This paper describes a new generic system for design and implementation of computerized protocols.

1. Introduction

Patient care is commonly managed by clinicians using general guidelines or specific protocols. Patients with acute, severe illness or injury may require care in an Intensive Care Unit (ICU), with many aspects of care managed concurrently. Variation of any aspect of care among patients, clinicians, ICUs and institutions is well known, and attributable to lack of standardization of clinical decision making, a recognized impediment to advancement of therapy and monitoring [1]. Automation of protocol management for specific aspects of care using computing technology and medical informatics/knowledge engineering principles is being addressed by several investigators. Computerization offers a consistent, systematic way to implement a protocol by prompting bedside clinicians in a timely manner, monitoring an individual patient's responses to therapeutic interventions, and monitoring the decision logic that is used to manage the aspect of care [1].

These protocols are complicated in nature, involving multiple variables measured at various time intervals (e.g., clinical laboratory and physiologic monitor) and multiple therapies, and requiring clinicians' (especially bedside nurses') constant observation, judgment, and accurate intervention. Comprehensive "paper" and computerized implementations of protocols based on current practice, literature evidence, and expert clinician opinion have been developed and shown to improve complex, long-term aspects of care compared to non-specific guidelines or ad hoc physician orders [2,3]. Compared to paper implementations, computerization offers much more consistent, systematic implementation, and comprehensive, prospective data acquisition.

2. Objectives

A general software framework and expert system tool were developed for the automation of ICU clinical care protocols. The design was based on an abstraction of current ICU protocols. An overriding goal of the design was to ensure that the system could represent any ICU protocol, and thereby to facilitate the implementation of any new protocol. There were two principal objectives of the project. First, a clinician (physician) should be able to "author" a protocol graphically, without requiring the assistance of a computer professional. This objective was met by developing a graphical flowchart-oriented expert system tool specifically designed for ICU protocols. Second, when a class of ICU protocols is encountered that requires capabilities not available in the current tools, a software developer should be able to add the needed capability with minimal effort and without a major rewrite of the system. This objective was met by implementing the tool using a framework of object classes that was tailored specifically for the ICU protocol domain, and at the same time employed good design practices to promote generality of application within the domain.

Table 1. Programmable logic modules

Object Class	PLM	Example
PlmFactInstruction	Boolean fact instruction	"Put the patient in a rotational surface bed."
PlmFactQuery	Boolean fact query	"Is the patient in a rotational surface bed?"
PlmMeasurementInstr	Measurement instruction	"Measure ICP"
PlmMeasurementQuery	Measurement query	"Is ICP > 20 mmHg?"
PlmMeasurementQuery2	Measurement query 2	"Was VE changed in the last 20 min?"
PlmMedInstr	Medication instruction	"Give morphine 10 mg now"
PlmMedQuery1	Medication query	"Was initial dose of morphine < 3 mg?"
PlmMedQuery2	Medication query 2	"Was morphine given in last 30 minutes?"
PlmMVMeasurementInstr	Multi-value measurement instruction	"Measure the patient's level of sedation."
PlmMVMeasurementQuery	Multi-value measurement query	"Is patient sedated to scale -2, -1, 0, 1, 2, 3, 4?"
PlmLabTestInstr	Lab test instruction	"Order this lab test: serum osmolality."

3. System design

In typical expert system development, there are four roles: 1) software developer - the developer of the expert system tool, 2) expert - who has experience and expertise in the problem domain, 3) knowledge engineer - who facilitates the encoding of the expert's knowledge and builds the expert system using the tool, and 4) user - of the completed expert system. In this project, a major goal was to provide a tool that was sufficiently convenient to use that the domain expert was able to construct the expert system without a knowledge engineer's help. This

combined role of expert and knowledge engineer were referred to as author. Therefore, we have three participants in this project: developer, author and user, as shown in Figure 1.

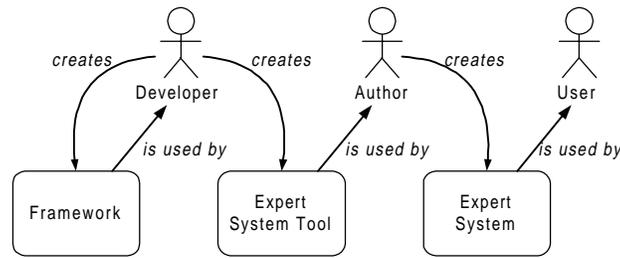


Figure 1. Three roles in protocol development

The user interface of the system is friendly and intuitive. The author of the protocols is supplied with a "canvas" on which the protocol flowchart can be drawn. The process of designing such a protocol is similar to what is commonly done on paper. The nurses and physicians in the ICU execute the protocol; a graphical message board supplies the user with instructions, and it also serves as a medium to input any feedback and treatment information.

4. Implementation

An object-oriented programming language (Java) was used to construct the framework, the expert system tool, and their principal elements. The components created in this way are modular and relatively independent of each other. The system was therefore designed to be platform-independent and deployable on most current platforms, and to be architecture neutral, not only at the source code level, but also at the binary level. The expert system tool can be executed on a standalone computer. If distribution of the tasks of the system is desired, good object-oriented design practice with clean interfaces between major components was employed so that migration to a distributed design over more than one computer communicating via the network will be relatively easy.

4.1. Authoring the flowchart

Based on the analysis of current ICU protocols, the system provides a variety of Protocol Logic Modules (PLM) to support five categories of facts: Boolean facts, single variable measurements, multi-value measurements, medications, and clinical laboratory analytical results. Fact Managers manipulate and organize the corresponding objects in these categories. Table 1 above shows the PLM modules currently provided along with some simple examples of each.

Each block in a protocol flowchart represents one of these PLMs. A simple example of a portion of a flowchart is illustrated in Figure 2, where the PLM type is written above each block.

After the author selects a PLM and inserts it into the flowchart, he must then "inspect" the block in order to specify details about the PLM. For example, after inserting the Measurement Query PLM in Figure 2, the author specified its details as shown in Figure 3.

In this fashion, by selecting PLMs, connecting them into the flowchart, and inspecting them to specify detailed instructions and queries, the author can create the entire protocol flowchart.

4.2. Executing the flowchart

The execution of the system is controlled by the Protocol Logic Controller (PLC). The PLC executes the flowchart by asking each PLM as it encounters it to execute. When a query PLM can respond based on information already known to the system, it does so without any user interaction. For instructions to the user (clinicians in the ICU), and for queries to which the user must respond, messages are sent to the Message Board, where a message can be selected for response by the user. When the user responds, either by acknowledging receipt/execution of an instruction, or by providing the data requested by a query, an ICU Event is returned to the corresponding Fact Manager to become part of the continually updated knowledge base.

The system time-stamps all ICU Events and manages timers for all of the facts. Therefore, as the PLC repeatedly executes the flowchart, a query such as "Has morphine been given in the last 30 minutes?" will eventually return a positive answer, resulting in the execution of a new path in the flowchart.

5. Feasibility testing and results

The system has been tested in two parts. An earlier version of the system, with similar architecture but different implementation features, was tested on patients in the ICU. The system described above has undergone preliminary testing, but has not been tested in the ICU.

The early prototype system was used to develop a computerized protocol for management of intracranial pressure (ICP) following traumatic brain injury. Development and bench testing was completed in electrical engineering and instrumentation labs using simulated patient monitors. Laboratory testing confirmed completeness of the knowledge base and appropriateness of responses to ICP and mean arterial pressure (MAP) variables. It also identified need to edit user-entered data, and for post hoc entry of therapy already administered to permit introduction of computerized protocol management at an arbitrary time in the clinical course of a particular patient. Clinical tests in an ICU proceeded with Institutional approval. The PC used to run the protocol was placed at bedside, and was used for monitoring but not for directing therapy.

Initial patient testing indicated no basic problems with the protocol development system or the concept of a computerized protocol for managing ICP. However, testing did highlight the widely recognized need for "clean" data in computerized decision support systems and several knowledge base problems unrelated to the decision support system tools. Subsequent patient testing demonstrated the advantage of a bedside computer processor to request necessary, timely clinical data and to comprehensively apply carefully developed clinical logic rules throughout a period of intense activity by the bedside nurse, physician, and other ICU personnel. The bedside clinicians found the computerized protocol to be very helpful in prioritizing care and providing precise nursing orders, and used ~70% of the computer-generated instructions during this first application of a computerized protocol in this aspect of intensive care.

Although the early prototype system worked well in the ICU, its implementation had shortcomings. The protocol logic was written in the CLIPS expert system language, which is known by few programmers and is difficult to modify. The graphical user interface was written in Objective-C on a NEXTSTEP system, and although it was easy to modify, it was not portable to other platforms. Experience with this first prototype led to the new system described above. The flowchart diagram design eliminated the difficulty with authoring and modification, and the Java implementation retained the ease of GUI design while providing portability. The new system has been tested successfully for basic functionality and ability to execute the flowchart, but has not yet been tested in the ICU.

6. Conclusion

The results of this project have successfully demonstrated the feasibility of automating ICU protocols using a computer system that provides both ease of authoring the protocol and a clinician-friendly user interface. The system has generality and should be useful in implementing a wide range of ICU protocols. As a specific example, a computerized protocol for management of ICP following traumatic brain injury was developed and tested in ICU environments. ICP management was shown to be a logical, patient-interactive process amenable to computerization. Rule-based, expert system principles were applied to successfully describe the ICP control process, and the feasibility of computer directed ICP management was demonstrated.

7. References

- [1] Morris AH. Algorithm based decision making. In: Tobin MJ, ed. Principles and Practice of Intensive Care Monitoring. New York:McGraw-Hill, 1997:1355-1381.
- [2] East TD, Heermann LK, Bradshaw RL, et al. Efficacy of computerized decision support for mechanical ventilation: Results of a prospective multi-center randomized trial. Journal of the American Medical Informatics Association 1999;6(Symposium Supplement 1999):251-255.
- [3] McKinley BA, Parmley CL, Tonnesen AS. Standardized management of intracranial pressure: A preliminary clinical trial. Journal of Trauma 1999;46(2):271-279.

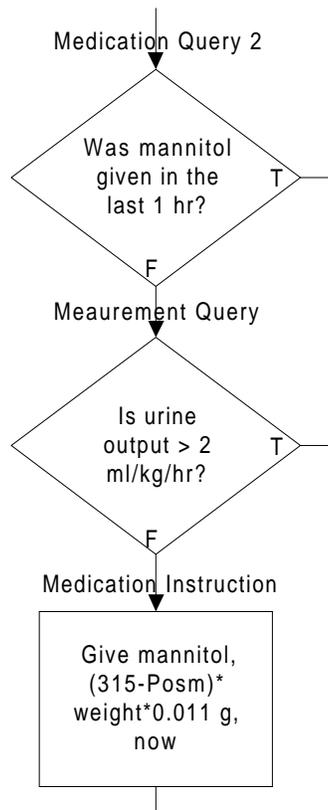


Figure 2. Simple flowchart showing connected PLMs

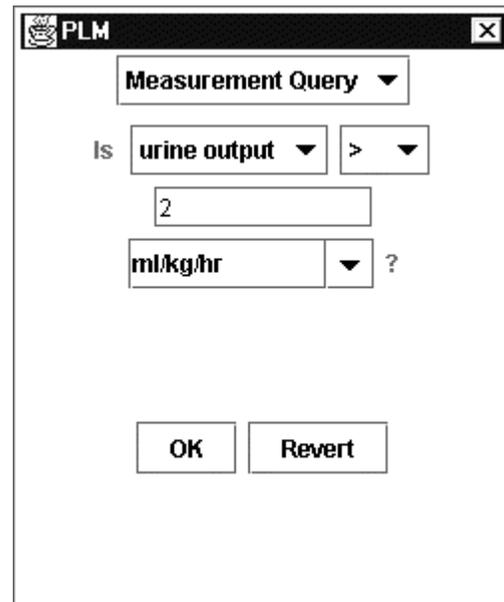


Figure 3. Inspection of a PLM