A Process to Maintain the Quality of a Computerized Knowledge Base

Gilad J. Kuperman, M.D., Ph.D., Julie M. Fiskio, Andrew Karson, M.D.
Partners HealthCare System, Department of Information Systems
Division of General Medicine, Brigham and Women's Hospital
Boston, MA

As part of a project to develop knowledge-based reminders for the outpatient setting, we developed a process to help maintain the quality of the knowledge base. The knowledge engineering process involved many parties, including several domain experts, a knowledge engineer, and a programmer and a process was necessary to assure that information transfer among individuals did not become confused. An MS Access database was created to store, among other data, textual versions of the rules as they evolved over time. In a 9-month period 36 rules were entered into the database. Of those, 17 are still active in their original form. The remaining 19 underwent various types of modifications; these changes were tracked in the database. Processes and tools to maintain knowledge bases are necessary if the benefits of clinical decision support systems are to be realized and investments in knowledge engineering are to be protected.

INTRODUCTION

The distinguishing characteristic of knowledge-based clinical decision support (KBCDS) software is that it encodes medical knowledge to allow the computer to "think like a clinician". KBCDS software can offer treatment suggestions or indicate where care is deviating from guidelines. Several KBCDS applications have been shown to improve the quality and lower the cost of health care.[1]

In a clinically active KBCDS application, the knowledge base does not stay static. Changes to the knowledge base are required as medical science evolves and subtle nuances of medicine are appreciated that were not obvious when the logic was created initially. For example, Jenders, et al.,[2] showed that over 78 months, 5500 changes were made to a set of 156 clinically active medical logic modules at Columbia-Presbyterian Medical Center; 39% of the changes affected the logic slot of the modules. Giuse, et al.,[3] found that when profiles of the Quick Medical Reference (QMR) diagnostic program were reviewed, 16% of the entries therein needed to be modified. A health care organization must be able to maintain the quality of a knowledge base over time.

Several kinds of functions to maintain a knowledge base are necessary. One important function is that non-programmer domain experts need to be able to review the contents of the knowledge base. Domain experts ultimately have responsibility for the content of the knowledge base and they must be able to assure that it is correct. For example, it should be easy to determine if the lower age limit for a cholesterol reminder rule is 20 or 25 years.

A second important maintenance function is the ability to know easily when a rule was changed. The date of a rule change is important when the impact of a rule is being evaluated; to measure the rule's impact, one must know when a particular version of a rule took effect. A third important function is the ability to detect easily what was the exact change between versions of a rule. For example, it may not be immediately evident simply by looking at procedural code or other knowledge representation that the lower age range of a cholesterol reminder has changed from 20 to 25; this fact should be highlighted so it is not overlooked if rule changes are reviewed. Also, the clinical rationale for any particular change should be easily evident so the evolution of the logic can be justified. A fourth function would allow the clinical body responsible for a rule to be identified.

Finally, knowledge engineering, the process of creating and updating an automated knowledge base, is a complex task. It may involve: 1) domain experts and/or clinical bodies that are the

1091—8280/99/$5.00

human repository of the medical knowledge, 2) a knowledge engineer who can model the medical knowledge in a computable form, and 3) a software developer whose job it is to encode the knowledge. The presence of several parties in the process creates a risk that the intent of the experts will be lost as the knowledge is transmitted among individuals, similar to the alteration of a message in the game of Telephone. The organization needs a process to ensure that the integrity of the knowledge is preserved.

As part of a project to create knowledge based reminders for physicians in the ambulatory setting, we developed a process to address the knowledge base maintenance functions mentioned above. This paper discusses our experiences with organizational issues as well as a tool we developed to help us with our task.

BACKGROUND

The setting for this project was Brigham and Women's Hospital (BWH), a tertiary care medical center in Boston, MA. Primary care practitioners (PCPs) at BWH deliver ambulatory services at a variety of on-site and off-site clinics. The Ambulatory Care Improvement Team (ACIT) at BWH* and the Information Systems Department at Partners HealthCare System† are involved in a project to deliver to PCPs patient-specific reminders intended to reduce the cost and improve the quality of care for ambulatory patients. These reminders are generated by the electronic medical record at BWH[4] and are printed on a clinical summary sheet used by the PCPs at the time of the patient visit. A synopsis of the outpatient reminder rules is shown in Table 1. The knowledge base is represented in the computer as discrete modules of procedural Mumps code, usually one module per rule.

* The Ambulatory Care Improvement Team is composed of PCPs from several clinics. The Team is charged with developing care improvement projects for the BWH ambulatory environment.

† Partners is the parent organization of BWH. Partners provides information services and other services to its component institutions.

Table 1. Outpatient reminder rules.

Rule category	Examples of rules
Health maintenance	Remind for: cholesterol every 5 years; pneumovax once for patients over 65; mammogram and pap smears annually in eligible women
Expensive medication reminders	Inform if there is a less expensive H2 blocker, HMG CoA reductase inhibitor, ACE inhibitor, or NSAID
Diabetic care	Remind for: annual ophtho exam; HbA1C every 6 months; urine protein annually;
Therapeutic recommendations	Inform if: MI and no ASA; MI and no beta blocker; diabetes and hypertension and no ACE inhibitor

METHODS

We developed a database to help assure the quality of the knowledge base for the reminders project. In this section, we will first describe the knowledge engineering process for the reminder project, because the database was created in part to solve problems of complexity in the knowledge engineering process.

The parties involved in the knowledge engineering process were (Figure 1): 1) a BWH general medicine faculty member who provided the starting set of rules, 2) the ACIT, which had the organizational responsibility for reviewing and approving the rules, 3) subspecialists who were consulted about difficult clinical issues as the rules were refined, 4) a general medicine fellow who led an evidence-based consensus development effort to finalize the rules, 5) a knowledge engineer in the IS department (trained in medical informatics) who worked with the general medicine fellow to assure that the rules were computable, and 6) a programmer who was responsible for implementing the rules.

Early in the project, as rules were being reviewed and refined, there was much confusion about the "current working version" of any

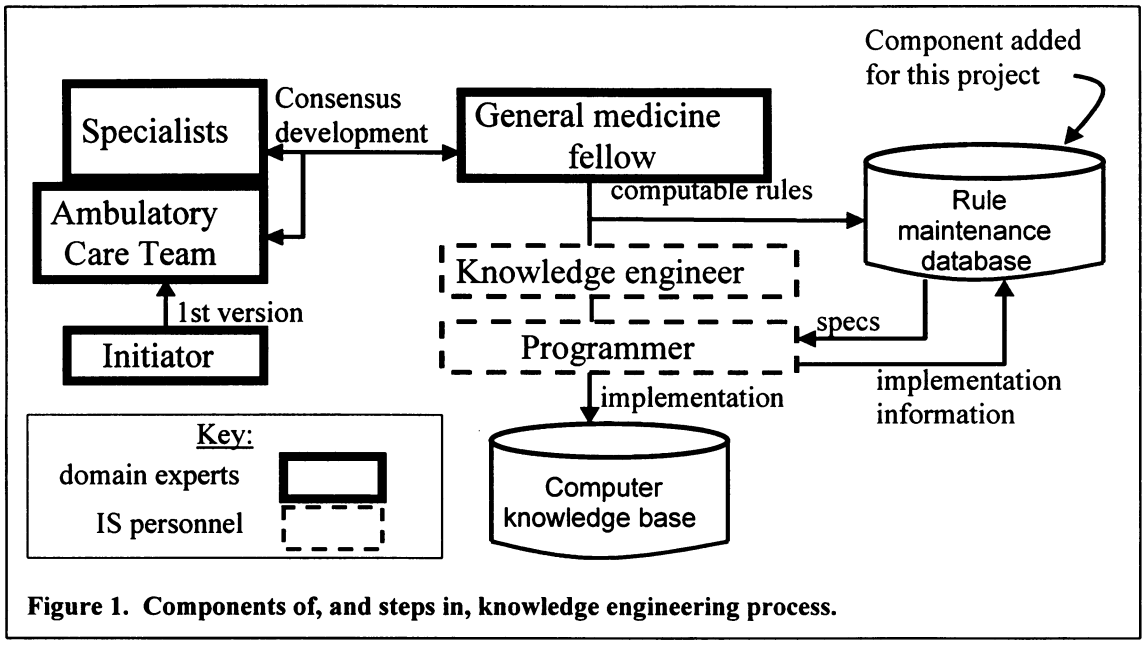

Figure 1. Components of, and steps in, knowledge engineering process.

particular rule. When modifications to a rule were suggested by one expert or another, it was unclear at what point the suggestion became "accepted". Also, when modifications were suggested serially in rapid succession, it was unclear which had successfully been communicated to the programmer. For these reasons, it became clear that it was necessary to have an "official" current working version of each rule. We created a database in MS Access to contain this and other relevant information. The goals of the database were: 1) to provide a human-readable version of the rules for the general medicine fellow to present to the other domain experts for comment, 2) to serve as a repository of which changes had been approved for implementation, 3) to highlight the clinical rationale for any particular change, 4) to serve as the specification for the programmer, and 5) to serve as a log for the programmer to document when specified changes had been implemented.

Use of the database (Figure 1)

The reminders project began in January, 1998. The database was created in April, 1998. From April forward, the general medicine fellow and the medical knowledge engineer would meet on a regular basis to review rules about which the domain experts had reached consensus. At this point, the rules would be entered into the database in a computable, albeit textual, format. Once in the database, the rule version was considered the "current working version". Copies of rules could be printed for the domain experts to review if questions of content arose. The knowledge engineer informed the programmer when changes had been made to the database. The programmer referred to the database when implementing the rules. After implementation was complete, the programmer updated the database with the date of implementation and the name of the Mumps routine that contained the knowledge.

The fields in the database are shown in Table 2. The "category" and "related rules" fields provide a 3-level ad hoc hierarchy by which rules were grouped. The "replaces" and "replaced by" fields provide pointers to previous and future versions of the rule. A rule that has been replaced by another rule is implicitly inactive; rules may be made explicitly inactive if there is no successor. An example of a rule that replaced another rule is shown in Table 2. A rule to suggest physicians that refer patients to a nurse for nutritional counseling if the most recent HbA1C level is greater than 8.0 was amended so the rule would be true only if the most recent HbA1C was within the last 6 months. The old version of the rule in Table 2 had modified an even older version in which the threshold for action was set at 8.9.

RESULTS

From 4/1/98 to 12/31/98, 36 initial versions of rules were entered into the database. Of these, 17 are still functioning with their original logic. The remaining 19 rules were altered for the following reasons: 6 rules related to reminding to check urine protein in diabetics were replaced with 7 new rules that incorporated enhanced logic; 2 rules related to checking cholesterol in diabetics were replaced with 4 rules that took into consideration the presence of coronary artery disease; 2 rules related to checking HDL in diabetics and one rule that suggested a nurse visit independent of HbAIC levels were inactivated; 5 rules related to checking cholesterol in patients with pre-existing CAD were inactivated in anticipation of a comprehensive cholesterol management algorithm being developed at our institution; and 3 rules relating to checking of cholesterol and HbA1C levels in diabetics were refined.

Table 2. Fields and two example records from rules maintenance database.

Field	Old version	New version
Rule title	Elevated HbA1C and no recent nursing visit	Elev. HbA1C and no recent nursing visit
Rule#	40	49
Active Flag	FALSE	TRUE
Category	Diabetes	Diabetes
Related rules	Nursing visit	Nursing visit
Logic	Diabetic, most recent HbA1C > 8.0, and no visit in last year w/ nurse as provider and diagnosis code = 250.0	Diabetic, HbA1C done in last 6 months and > 8.0, and no visit in last year w/ nurse as provider and diagnosis code = 250.0
Displayed message	Elevated HbA1C (xx.x). Suggest nursing education visit.	Elevated HbA1C (xx.x). Suggest nursing education visit.
Rationale for rule/change	Threshold changed to 8.0 based on conversations b/w Andy Karson and diabetic specialists.	Add clause to make sure HbA1C done in last 6 months
References		
Approved by	Amb CIT	Amb CIT
Approved	7/15/98	9/3/98
Routine	REMDM	REMDM2
Implemented	7/30/98	9/28/98
Text comments	Diabetic defined as "diabetes flag" in Mini-Amb	Diabetic defined as "diabetes flag" in Mini-Amb
Replaces	23	40
Replaced by	49	

DISCUSSION

We have described a process and a database to support the maintenance of a computerized knowledge base that is part of a clinical KBCDS application. The project formalized what would have been otherwise an informal and unmanaged process. We believe that processes and tools such as these will be needed increasingly as more institutions use KBCDS applications to realize the potential benefits of computerized clinical information systems. As we have described, the process of creating a complex knowledge base can be quite unwieldy. As AMIA has suggested,[5] institutions should demonstrate they have in place processes to handle such complexity. Also, a clinically active automated knowledge base represents a significant investment of time and effort by several individuals. A knowledge base that is not maintained will become out of date, the KBCDS application will not function as it should, and the institution's initial investment in knowledge engineering will have been wasted.

It is important to note that the process we have described is not intended to assure that the software itself functions as it is specified. Such assurance would require a separate testing and general software QA effort. However, our process does help assure that the intent of the domain experts has been represented as well as possible and that the specifications are then explicitly documented. The specifications could then be used as a basis for further software QA. There are various ways to implement knowledge that ease the actual verification. The Arden Syntax[6] is one method to represent a knowledge base in an easily maintainable manner but there are others.[7]

The Arden Syntax[6] is a standard for the representation for medical knowledge. An institution that uses the Arden Syntax might be

able to use the repository of logic modules to perform some of the functions performed by our rule maintenance database. For example, Arden (although a procedural programming language) is relatively readable; the logic modules themselves could serve as the readable version of the rules. Also, because the logic is represented as text modules, saving previous versions of the modules can assist version control.[2] The library section of the modules can be used to document the intent of any particular version. However, the availability of the tools does not guarantee their use; the institution must have in place a process that assures these tools will be used properly.

Several domain experts participated on the project described in this paper. One might comment that fewer domain experts might have eased the knowledge maintenance problem because the cross traffic of information would have been less. We believe that increasingly, knowledge bases will be scrutinized by larger numbers of domain experts, either because the medicine behind the logic will be very complex, or because the political landscape demands that several parties review and comment on the rules as part of the buy-in process. We believe that the days of one expert at an institution creating a knowledge base are numbered.

Ideally, a database such as we have described here would be an "active" database that sends a message to the programmer when the active flag on a rule is changed. Similarly, the knowledge engineer could receive a message if the implementation information is not updated within a short time after an active flag is modified.

CONCLUSION

We have described an institutional process and a database to help to assure the quality of an outpatient reminder knowledge base. Without formal processes and tools, the state of the knowledge base could be hard to discern, managing changes to the knowledge base would be difficult, and the risk of errors in the knowledge base would be high. As complex knowledge becomes a more common component of clinical information systems, formal processes for managing knowledge bases will become increasingly important .

ACKNOWLEDGEMENTS

This work was supported by a grant from the Aetna-U.S. Healthcare Academic Medicine Managed Care Initiative.

REFERENCES:

[1] Hunt DL, Haynes RB, Hanna SE, Smith K. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review. JAMA 1998 Oct 21;280(15):1339-46.

[2] Jenders RA, Huang H, Hripcsak G, Clayton PD. Evolution of a knowledge base for a clinical decision support system encoded in the Arden Syntax. Proc Amia Symp 1998;:558-62.

[3] Giuse DA, Giuse NB, Miller RA. Evaluation of long-term maintenance of a large medical knowledge base. J Am Med Inform Assoc 1995 Sep-Oct;2(5):297-306.

[4] Teich JM, Geisler MA, Cimerman DE, Frank AD, Glaser JP. Design considerations in the BWH ambulatory medical record: features for maximum acceptance by clinicians. AMIA (SCAMC) Proc 1990;:735-739.

[5] Miller RA, Gardner RM. Recommendations for responsible monitoring and regulation of clinical software systems. American Medical Informatics Association, The Computer-based Patient Record Institute, The Medical Library Association, The Association of Academic Health Science Libraries, The American Health Information Management Association, and The American Nurses Association. Ann Intern Med 1997 Nov 1;127(9):842-5.

[6] Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res 1994 Aug;27(4):291-324.

[7] Kuperman GJ, Teich JM, Bates DW, et al. Representing hospital events as complex conditionals . AMIA (SCAMC) Proc 1995;:137-141.