Desperately Seeking Data: Knowledge Base-Database Links

George Hripcsak, M.D.; Stephen B. Johnson, Ph.D.; Paul D. Clayton, Ph.D.
Center for Medical Informatics
Columbia-Presbyterian Medical Center, New York, NY 10032

Linking a knowledge-based system (KBS) to a clinical database is a difficult task, but critical if such systems are to achieve widespread use. The Columbia-Presbyterian Medical Center's clinical event monitor provides alerts, interpretations, research screening, and quality assurance functions for the center. Its knowledge base consists of Arden Syntax Medical Logic Modules (MLMs). The knowledge base was analyzed in order to quantify the use and impact of KBS-database links. The MLM data slot, which contains the definition of these links, had almost as many statements (5.8 vs. 8.8, ns with p=0.15) and more tokens (122 vs. 76, p=0.037) than the logic slot, which contains the actual medical knowledge. The data slot underwent about twice as many modifications over time as the logic slot (3.0 vs. 1.6 modifications/version, p=0.010). Database queries and updates accounted for 97.2% of the MLM's total elapsed execution time. Thus, KBS-database links consume substantial resources in an MLM knowledge base, in terms of coding, maintenance, and performance.

INTRODUCTION

A major challenge to using knowledge-based systems (KBSs) in clinical care is the linking of the KBS to a real clinical database. Despite the multitude of systems that have been reported in the literature over 30 years, and despite the number of evaluations showing that KBSs ought to be useful [1-4], a tour of the average hospital would convince one that these systems have little use in clinical care. Basic KBS research may reduce this discrepancy — through better completeness, better explanations, new adaptable models, and so on — but it will not eliminate it. Part of the problem lies not within the KBS itself, but in the link between the KBS and the clinical environment [5-7]. The routine availability of more and more clinical data in coded electronic form is reducing one of the hurdles to using KBSs: the manual entry (and re-entry) of clinical data. For example, the authors of QMR report that one of the factors discouraging the use of QMR is the manual entry of data [8]. To reap this benefit, however, a major effort is required to map conceptual entities in the KBS to actual entries in the clinical database.

The process of linking a KBS entity to a clinical database entry is a complex one. The first step in creating a link is to define a database query (links also include database updates, but they tend to be less problematic). The desired KBS entity must be specified, the entries available in the clinical database must be reviewed, the appropriate terms and retrieval methods must be chosen, and a syntactically correct query must be assembled. This requires knowledge of the organization of the database, the terminology used to store the information, and the syntax of the query. At Columbia-Presbyterian Medical Center (CPMC) most data are stored centrally in a relational patient database [9]. Admit-discharge-transfer information are stored in a older hierarchical database, and free text data are stored in indexed files. The terminology is represented in a semantic network known as the Medical Entities Dictionary [10]. The CPMC knowledge base consists of independent rules called Medical Logic Modules (MLMs) written in the Arden Syntax [11] and executed by the clinical event monitor [12]. MLMs have been used to provide medical alerts, interpretations, clinical research screens, and quality assurance functions. Queries are defined using the Arden Syntax along with CPMC-specific constructs. (The Arden Syntax only defines part of a query; due to differences among institutions, the rest of the query is defined locally [13].)

The second step is to characterize the data that are returned and develop the filtering necessary to convert the raw data into a form that meets the needs of the KBS. Even when the basic query is written, a naive attempt to use raw data may lead to erroneous results. This is because KBSs like QMR [14] and DXplain [15] exploit the preprocessing and filtering that clinical users do naturally. A QMR finding like "potassium serum increased" assumes that the clinical user has already determined that the laboratory value is recent enough (not 10 years old), that it is not an obvious laboratory error (100 mEq/dl), that it is not a special value (Quantity Not Sufficient), and that it is increased above normal for the laboratory. If the KBS is linked directly to a database, the computer must assume the task of filtering the raw data.

The third step is to link the query to the KBS. In the case of the Arden Syntax, a query can be inserted directly into an MLM. In the case of a KBS like

QMR, an external facility needs to be written to provide the KBS with the query results. Wherever possible, queries that are already linked to the KBS should be reused.

The fourth step is to test the query within the environment of the KBS. This can be the most time consuming step. The best way to test an MLM is to turn it on (i.e., let it run in real time), but send the generated messages to the MLM author instead of the patients' clinicians. The author can review the messages to see whether they were appropriate. The difficulty with this approach is that most MLMs fire rarely, so this sort of test can take two or more weeks to gather sufficient information. MLMs are refined based on the results of the test, and then they are retested. The entire process can take months.

At CPMC, five years have been invested in building a clinical information system with automated decision support [12]. We have found that the greatest challenge to effective decision support is getting the data. Even when the data are available in the clinical database, finding where those data are stored and converting those data into a form acceptable to the KBS requires extensive knowledge of the database and the vocabulary. The CPMC experience has been that the writing and testing of queries consumes more time than all the other KBS tasks combined. In order to quantify the use and impact of KBS-database links on the CPMC knowledge base, three aspects of MLMs were evaluated: the amount of MLM code dedicated to links, the number of modifications to links, and the proportion of execution time spent on links.

METHODS

Code Size

This evaluation was based upon the MLM's organization of slots [11]. Four MLM slots were analyzed in detail. (1) The **data** slot defines the mapping from MLM entities (local variables) to entries in the patient database and the institutional vocabulary. It includes queries to retrieve data ("read" statements), definitions of events like database updates that trigger the MLM ("event" statements), and specifications of how MLM messages are to be stored in the patient database ("message" and "destination" statements). By far, the bulk of the data slot is dedicated to queries. (2) The **evoke** slot uses events defined in the data slot to specify time constraints on how the MLM is to be triggered (immediately, after a time delay, or periodically). (3) The **logic** slot first filters the raw data retrieved by the data slot's queries. It then tests a set of criteria,

```
DATA:
 /* HL7 A01, A06 =  admission */
 admission := event {
   '32511~management event',
   '32467~admit a patient'};

 /* check that admission is inpatient */
 inpt_case := read last
   {'evoking', 'dam'="gydapmp",
   'constraints'=" I***"; "hcase"; "K"};

 /* patient birthdate */
 birthdate := read last
   {'dam'="gydapmp";"hpbasic";"hbirthdt"};

 /* get medical service for admission */
 service := read last
   {'evoking', 'dam'="gydapmp";
   "hcasex"; "hadmsvcl"};

 /* get mrn of mother of newborn */
 mother := read last
   {'evoking', 'dam'="gydapmp";
   "hnwbmthr"; "hnwbmrn"};

 /* hepatitis B surf antigen of mother */
 mom_hbsag := read last 3 from
   {'mrn'=mother, 'dam'="pdqresl";
   '1900~hepatitis serology panel',
   '2210~hepatitis B surface antigen';
   '1494~blood hepatitis B surface antigen'};
 ;;
EVOKE:       admission;;
LOGIC:
 /* exit if not inpatient or no mother */
 if inpt_case is null or mother is null then
   conclude false;
 endif;

 /* define newborn  */
 age := int((eventtime-birthdate)/1 day);
 if service="NUR"
 or age is within 0 to 1 then

   /* look for pos hep B surface Ag */
   pos_hbsag := last(mom_hbsag where
     (it <> "NEG" and it <> "QNS"));
   if pos_hbsag is present then
     conclude true;
   endif;

 endif;
 ;;
ACTION:
 write "This newborn's mother (" ||
   mother|| ") had a hepatitis B surface
   antigen test of" ||pos_hbsag|| " on " ||
   time of pos_hbsag|| ", and the newborn
   must be treated appropriately for this
   result.";
 ;;
```

Figure 1. *This MLM, called* ***newborn_hepatitis_B****, sends an alert to a physician if a newborn's mother has an active hepatitis B infection, since the newborn must be treated for the condition. Only the four main knowledge slots are shown here: data, evoke, logic, and action.*

performs a trend analysis, or executes an algorithm. If its conclusion is "true," then the action slot is executed. (4) The **action** slot defines the wording of the MLM message, and uses the data slot's definitions to store the message in the database or send the message via electronic mail. Thus, the data slot contains the definitions to link the MLM to the patient database (KBS-database links), whereas the logic slot contains the bulk of the actual medical knowledge coded in the MLM. Figure 1 shows an example.

The MLMs that are currently active at CPMC were analyzed. Two measures of the amount of code per slot were used: number of statements and number of tokens. The latter was included because a single data slot statement tends to be longer and to require more effort than a single logic statement. A "token" is defined in ASTM Standard Specification E1460 [11]; it includes numbers, times, terms, strings, identifiers, and operators. Comments are explicitly excluded.

Modifications

Every time an author edits an MLM, its version number is incremented. Consecutive versions of MLMs were analyzed to determine how many modifications were made in each MLM slot. The Unix **diff** utility [16] was used to count how many blocks of code were altered between versions. It compares two files line by line, grouping consecutive altered lines into a single block. The **diff** utility provided an efficient and objective measure of modifications.

Execution Time

After compilation, MLMs are executed by the clinical event monitor on an IBM mainframe computer (3090 Model 300). The compiled MLMs were divided into three components: queries to the patient database, updates to the patient database, and MLM logic. On each MLM evocation, the elapsed time was measured for each component. The queries and updates included only the time spent actually communicating with the patient database. Setting up query parameters and assembling MLM messages was included in the MLM component. Electronic mail routing is performed asynchronously, so it appears to take a negligible amount of time; the little time that did register was counted with the updates.

RESULTS

Code Size

The CPMC MLM knowledge base has been running for research purposes since September 1990 and for clinical care since March 1992. Sixty unique MLMs have been written and used, and 20 are active at the current time (most of the others were short-term research screening MLMs).

The mean and sample standard deviation of the number of statements and tokens for each of the slots are shown in Table 1. The data slot had fewer statements than the logic slot (not significant, $p=0.15$), but more than the other slots ($p<0.001$). The data slot had more tokens than the other three slots (vs. logic, $p=0.037$; vs. evoke and action, $p<0.001$).

Table 1. Code per MLM slot.

	statements		tokens		tokens/
slot	mean	stddev	mean	stddev	stmnt
data	5.8	2.0	122	62	21
evoke	1	0	6	6	6
logic	8.8	9.0	76	70	9
action	1.6	0.8	61	43	38

Modifications

A log was available on 20 MLMs since as early as September 1990. In these MLMs, there were a total of 113 versions, or 93 comparisons between versions. Table 2 shows the total number of modifications in each slot, the mean number of modifications between consecutive versions, and the sample standard deviation. The data slot had significantly more modifications than the other slots (vs. logic, $p=0.003$; vs. evoke and action, $p<0.001$). The mean time between versions was 50 days with a range of 0 to 765 days.

Table 2. Modifications per MLM slot.

slot	total modifications	modifications/ version	sample stddev
data	183	1.97	1.99
evoke	18	0.19	0.61
logic	99	1.06	2.18
action	97	1.04	1.12

Execution Time

The mean and median elapsed time for the three MLM components are shown in Table 3. All times are in milliseconds. The mean number of queries per MLM evocation was 1.16, and the mean number of messages per evocation was 0.087. MLMs were evoked at a rate of about 0.84 per second. MLM logic represented only 2.8% of the elapsed time.

Table 3. Execution time (elapsed millisec).

component	mean	median
query	194.9	36
update	96.1	0
logic	8.4	2

DISCUSSION

These results demonstrate that at CPMC, KBS-database links use the most code (measured in tokens), require the most maintenance (measured in MLM modifications), and consume the greatest amount of execution time. The larger number of modifications may simply reflect the larger amount of code dedicated to queries. Admittedly, the definition of what comprises a KBS-database link and what comprises MLM logic is vague. One could argue that the action slot, which defines the text of the message to be stored in the database, could be counted with either the data slot or the logic slot. Much of the code necessary to filter raw data (e.g., eliminate "null" potassium values) was included in the logic slot in this analysis; one could argue that it should be included with the data slot since it is dependent on how the data are stored in the database. Therefore this analysis may be a conservative measure of the importance of KBS-database links in MLMs.

The CPMC performance results (Table 3) are interesting because they reflect a real KBS working with a real patient database with 10 gigabytes of data. The fact that elapsed time was used tends to overemphasize the effect of database access on overall throughput. Empirically it was determined that as much as 80% of query and update time may be due to wait periods. Even so, the logic would still represent only 13% of total CPU throughput. The large disparity between the mean and median query times reflects the wide variation in database query complexity. A few queries, for example, actually parse a collection of free text reports in real time; when the system is busy, they can take 20 seconds to parse a large number of reports. Most MLM evocations do not result in the generation of MLM messages. Usually, an MLM is able to conclude "false" and terminate after reading only a subset of its data. This is why the median update time is zero and the mean number of queries per evocation is low. More elaborate studies of system performance have been published [17], but simulations were used.

The Arden Syntax is designed for small, independent rules, most of which are data-driven. Naturally, one would expect the KBS-database link to be a large part of their creation and maintenance. When a KBS with a more elaborate reasoning strategy (expert system, belief network, ...) is linked to a real clinical database, the proportion of code and effort dedicated to the KBS-database link may be higher or lower than the MLMs. While the actual knowledge content may be greater and more complex, the KBS-database link will also be more extensive and more complex. For example, linking QMR's [14] 4200 findings and 600 diseases to a real clinical database would take an enormous effort. No matter what type of KBS is used, the KBS-database link will be a critical and difficult component.

Since no single institution will ever create a complete knowledge base, knowledge must be shared among institutions [18]. Reviews of several well-described clinical information systems reveal that there is little correspondence among the data models and vocabularies [13,19], and a KBS created at an institution will reflect the local data model and vocabulary. Thus, even if it is possible to share the medical knowledge, the link to the patient database will have to be redone for each institution. These links may become the bottleneck that prevents the benefits of sharing from being fully realized.

Tools can be used to ease the task of creating and maintaining KBS-database links. Some tools help clinicians and researchers review data in a patient database [20-22]. Other tools have been explicitly designed for testing a knowledge base using real data and extracting information automatically [23]. Vocabulary tools define mappings from local terms to entries in a database [24-25]. The UMLS Patient Database project [26] uses declarative frames to map from a variety of patient databases to a single common database.

At CPMC, no simple answer has been found to reduce the work of generating and maintaining KBS-database links. The best solution that we have found has been to provide a working environment rather than a single tool. The environment currently includes the Medical Entities Dictionary to select vocabulary terms, provisions to reuse queries via an MLM editor [27] (now implemented using simple cutting and pasting), facilities to run queries interactively to review their results, and mechanisms to run MLMs against the patient database for the purpose of debugging them. As patients, coding schemes, and systems change, so must the KBS-database links change. The CPMC Automated Statistical Tracker [28] monitors the activity of each MLM individually, looking for statistically significant changes in its rate of message generation. When such a case is found, the MLM is investigated for malfunctions, and occasionally a modification needs to be made to a query.

CONCLUSIONS

The KBS-database link is a significant and costly part of the CPMC knowledge base, measured in terms of MLM coding, MLM maintenance, and MLM execution. CPMC has invested in an environment to create and maintain these links, but the effort remains substantial.

Acknowledgement

This work was supported by the International Business Machines Corporation and by a grant from the National Library of Medicine LM04419 (IAIMS).

References

[1] Evans RS, Larsen RA, Burke JP, et al. Computer surveillance of hospital-acquired infections and antibiotic use. JAMA 1986;256:1007-11.

[2] McDonald CJ, Hui SL, Smith DM, et al. Reminders to physicians from an introspective computer medical record. Ann Intern Med 1984;100:130-8.

[3] Barnett GO, Winickoff RN, Morgan MM, Zielstorff RD. A computer-based monitoring system for follow-up of elevated blood pressure. Med Care 1983;21:400-9.

[4] Rind DM, Safran C, Phillips RS, et al. The effect of computer-based reminders on the mangement of hospitalized patients with worsening renal function. In: Clayton PD, ed. Proc SCAMC 15, 1992; 28-32.

[5] Shortliffe EH. Computer programs to support clinical decision making. JAMA 1987;258(1):61-6.

[6] Wyatt J. Computer-based knowledge systems. Lancet 1991;338:1431-6.

[7] Miller RA. INTERNIST-1/CADUCEUS: problems facing expert consultant programs. Meth Inform Med 1984;23:9-14.

[8] Miller RA, Masarie FE. The demise of the "Greek Oracle" model for medical diagnostic systems. Meth Inform Med 1990;29:1-2.

[9] Johnson S, Friedman C, Cimino JJ, et al. Conceptual data model for a central patient database. In: Clayton PD, ed. Proc SCAMC 15, 1992; 381-5.

[10] Cimino JJ, Hripcsak G, Johnson SB, Clayton PD. Designing an Introspective, Multipurpose, Controlled Medical Vocabulary. In: Kingsland LC, ed. Proc SCAMC 13, 1989; 513-8.

[11] ASTM. E 1460 Standard Specification for Defining And Sharing Modular Health Knowledge Bases (Arden Syntax for Medical Logic Modules). ASTM Standards, v 14.01. Philadelphia: ASTM, 1992; 539-87.

[12] Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center decision-support system as a model for implementing the Arden Syntax. In: Clayton PD, ed. Proc SCAMC 15, 1992; 248-52.

[13] Hripcsak G, Clayton PD, Pryor TA, Haug P, Wigertz OB, van der Lei J. The Arden Syntax for Medical Logic Modules. In: Miller RA, ed. Proc SCAMC 14, 1990; 200-4.

[14] Miller RA, McNeil MA, Challinor SM, Masarie FE, Myers JD. The INTERNIST-1/Quick Medical Reference report. West J Med 1986;145:816-22.

[15] Barnett GO, Cimino JJ, Hupp JA, Hoffer EP. DXplain: an evolving diagnostic decision-support system. JAMA 1987;258:67-74.

[16] Rosen KH, Rosinski RR, Farber JM. UNIX System V Release 4. Berkeley: McGraw-Hill, 1990.

[17] Haimowitz IJ, Kohane IS. Influences on the performance of hospital clinical event monitoring. In: Clayton PD, ed. Proc SCAMC 15, 1992; 614-8.

[18] Clayton PD, Pryor TA, Wigertz OB, Hripcsak GM. Issues and structures for sharing knowledge among decision-making systems: The 1989 Arden Homestead Retreat. In: Kingsland LC, ed. Proc SCAMC 13, 1989; 116-21.

[19] Stead WW, Wiederhold G, Gardner R, Hammond WE, Margolies D. Database systems for computer-based patient records. In: Ball MJ, Collen MF, eds. Aspects of the Computer-based Patient Record. New York: Springer-Verlag, 1992.

[20] Safran C, Porter D, Lightfoot J, Rury CD, Underhill LH, Bleich HL, Slack WV. ClinQuery: a system for online searching of data in a teaching hospital. Ann Intern Med 1989;111(9):751-6.

[21] Hammond WE, Straube MJ, Blunden PB, Stead WW. Query: the language of databases. In: Kingsland LC, ed. Proc SCAMC 13, 1989; 419-23.

[22] Morgan MM, Beaman PD, Shusman DJ, Hupp JA, Zielstorff RD, Barnett GO. Medical Query Language. In: Heffernan HG, ed. Proc SCAMC 5, 1981; 322-5.

[23] Haug P, Hoak S. Veristat: a support tool for knowledge development. In: Miller RA, ed. Proc SCAMC 14, 1990; 650-4.

[24] Timmers T, van Mulligen EM, van den Heuvel F. Integrating clinical databases in a medical workstation using knowledge-based modeling. In: Lun KC, et al, eds. Proc MEDINFO 92, 1992; 478-82.

[25] Annevelink J, Young CY, Tang PC. Heterogenous database integration in a Physician Workstation. In: Clayton PD, ed. Proc SCAMC 15, 1992; 368-72.

[26] Fu LS, Bouhaddou O, Huff SM, Sorenson DK, Warner HR. Toward a public domain UMLS patient database. In: Miller RA, ed. Proc SCAMC 14, 1990; 170-4.

[27] Gao X, Shahsavar N, Arkad K, Ahlfeldt H, Hripcsak G, Wigertz O. Design and functions of medical knowledge editors for the Arden Syntax. In: Lun KC, et al, eds. Proc MEDINFO 92, 1992; 472-7.

[28] Hripcsak G. Monitoring the monitor: automated statistical tracking of a clinical event monitor. Comput Biomed Res, in press.