Representing Hospital Events as Complex Conditionals

Gilad J. Kuperman, M.D., Ph.D., Jonathan M. Teich, M.D., Ph.D., David W. Bates, M.D., M.Sc.
John McLatchey, B.A., Thomas G. Hoff, B.A.
Center for Applied Medical Information Systems Research, Brigham and Women's Hospital, Boston, Massachusetts.

ABSTRACT

We have developed an approach to medical knowledge representation whereby simple medical concepts are combined to yield complex statements of testable medical logic. The logic is created from a small number of generic medical concepts that are instantiated and combined to create the rules. Rule writing is done through a rule editor and requires knowledge of the system's data dictionaries, though no programming is required. We have used the approach to create a large knowledge base including panic lab alerting rules, drug-laboratory interaction alerting rules, an adverse drug event monitor, and a drug-age interaction detection program. The rules have been used as part of an alerting system and for data collection to determine the frequency of events of interest. The scheme is extensible and yields a readable form of the created knowledge. The scheme holds great promise as a durable form of medical knowledge representation.

INTRODUCTION

Although computerized clinical information systems (CISs) are becoming ubiquitous throughout the nation's medical institutions, the presence of decision support features[1] within these systems remains uncommon. Many of the issues that arise when implementing decision support stem from the characteristics of the CIS itself (e.g., available processing power, user interface, coded data, e-mail, interface to paging systems, automated coverage lists, etc.). An additional important issue is the manner in which the medical knowledge is represented within the electronic knowledge base.[2] Medical knowledge is most often represented procedurally using the CIS's native programming language. Procedural medical knowledge can be robust (since it is described by a general programming language) and requires no additional system software tools. However, using the CIS's native programming language does nothing to ease the "knowledge acquisition bottleneck"[3] because knowledge base creation and maintainence requires knowledge of the CIS's programming environment. To ease the knowledge maintenance task, some researchers have developed programming languages specifically intended to represent medical knowledge.[4],[5],[6] These languages are designed to be easily readable and to optimally manipulate medical concepts however knowledge creation and editing requires knowledge of the language's syntax. Also, special compilers are required to convert the rules from the special language into the native language of the CIS. At our institution, we have had success with a declarative, "slot-filling", approach to representing medical knowledge, specifically in a drug-drug interaction detection application.[7] Others have also used declarative representations of knowledge for decision support and other projects.[8],[9] The goal of the current project was to represent a more general class of hospital events declaratively and to evaluate this representation for alerting and data collection purposes.

METHODS

General considerations: We had observed that Tate[10] and Rind[11] had developed rule-based alerting systems that had been demonstrated to have a positive impact on patient care. We therefore felt that, at a minimum, our structure should be able to emulate their rules. We observed that each of their rules could be described by boolean combinations of simpler medical conditions. Tate's rules are shown in Table 1. For example, Tate defines hypokalemia to be present when:

[K+ < 2.7]
$\vee$ ([K+ < 3.3] $\wedge$ [patient is on digoxin])
$\vee$ ([K+ < 3.2] $\wedge$ [K+ has fallen more than 1 over 24 hours]).

($\vee$ represents logical 'or' and $\wedge$ represents logical 'and'.)

Rind defines worsening renal function as:

[creatinine rises 0.5 while the patient is on a nephrotoxic medication]
$\vee$ ([creatinine rises 50% while the patient is on a renally excreted medication] $\wedge$ [creatinine > 2.0]).

We further noted that many of the simpler medical

Table 1. Tate's life-threatening laboratory alert rules.

Hyponatremia	Na < 120
Falling sodium	Na fell 15 in 24 hours and Na < 130
Hypokalemia	K+ < 2.7 or K+ < 3.3 and patient on digoxin
Falling potassium	K+ < 3.2 and K+ fell 1 over 24 hours
Metabolic acidosis	
	CO2<15 and BUN>50 or CO2<18 and BUN<50 or CO2<18 or CO2<25 and fallen 10 in 24 hours
Falling Hematocrit	Hct fallen 10 since last result and Hct<35 or Hct fallen 5 since last result and Hct fallen .4/hr since last result or Hct fallen 5 since last result and Hct <16

conditions were simply different instantiations of the same medical concept. For example, 'Creatinine > 2.0' and 'K+ < 2.7' could both be represented by the same 3-slot generic concept: 1) a laboratory test, 2) greater than or less than, and 3) a numerical value. Similarly, 'Na fell 15 over 24 hours' and 'K+ fell 1 over 24 hours' could both be represented by a 4-slot concept: 1) a laboratory test name, 2) a directional change ('rise' or 'fall'), 3) a numeric value, and 4) a time interval. We termed such multi-slot generic concepts "primitives".

We hypothesized 1) that many events of medical interest could be represented by boolean combinations of simple medical conditions, and 2) that many simple conditions could be represented by different instances of the same generic concept (primitive). We chose to represent medical events as boolean combination of conditions of the form:

$$\begin{array}{ll} & (\alpha_{11} \wedge \alpha_{12} \wedge \ldots) \\ \vee & (\alpha_{21} \wedge \alpha_{22} \wedge \ldots) \\ \vee & (\alpha_{31} \wedge \alpha_{32} \wedge \ldots) \\ & \vdots \end{array}$$

Each of the α_{ij}s is called a "condition" and is an instantiation of a primitive. Each complete expression therefore is a "list of lists of conditions".

We chose to represent events as a list of lists of conditions rather than as a general (i.e., unrestricted) boolean combination of conditions because it was easier to build an editor for the more restricted form and we felt even the restricted form could express a large class of medical events.

Setting: The work was carried out at Brigham and Women's Hospital (BWH), a 726-bed tertiary care hospital in Boston, MA. Computing services at the hospital are provided by the Brigham Integrated Computing System (BICS), an internally developed personal computer (PC) local-area-network (LAN) based hospital information system providing administrative, financial, and clinical computing services. Applications software is written in Mumps and data (or, in this case, knowledge) are stored in a distributed Mumps database. We used the above stated knowledge representation scheme in a general purpose alerting and data collection program we are developing.

RESULTS

Primitives developed to date: As of this writing (April 1995), we have defined 15 primitive medical conditions (Table 2) that are available for rule writing. For each primitive, the parameters that need to be instantiated to create a condition are shown in capital letters. For example, in primitive #1, MEDICATION is a field (slot) to be filled in to instantiate the primitive and make it a condition. The value of MEDICATION would be selected from the order entry medication data dictionary used in the computerized order entry application.[12] In primitive #3, the fields LAB, RISEN/FALLEN,

Table 2. List of primitives available for rule writing.

1. patient on MEDICATION
2. Any of last L LAB results within last M days are OP VALUE
3. LAB has RISEN/FALLEN VALUE over K hours
4. LAB has RISEN/FALLEN VALUE since previous result over last M days
5. LAB has RISEN/FALLEN VALUE per hour since previous result
6. LAB has RISEN/FALLEN K percent of previous value
7. Prior LAB Result within last M days is OP VALUE
8. FIELD is OP VALUE
9. patient on MEDICATION over last M days
10. MEDICATION previously ordered over last M days
11. TRUE
12. patient has GFR<15
13. patient had LAB ordered over last M days
14. patient had a TPN order over the last M days
15. LAB has RISEN/FALLEN by VALUE/PERCENT while on MED in last M days.

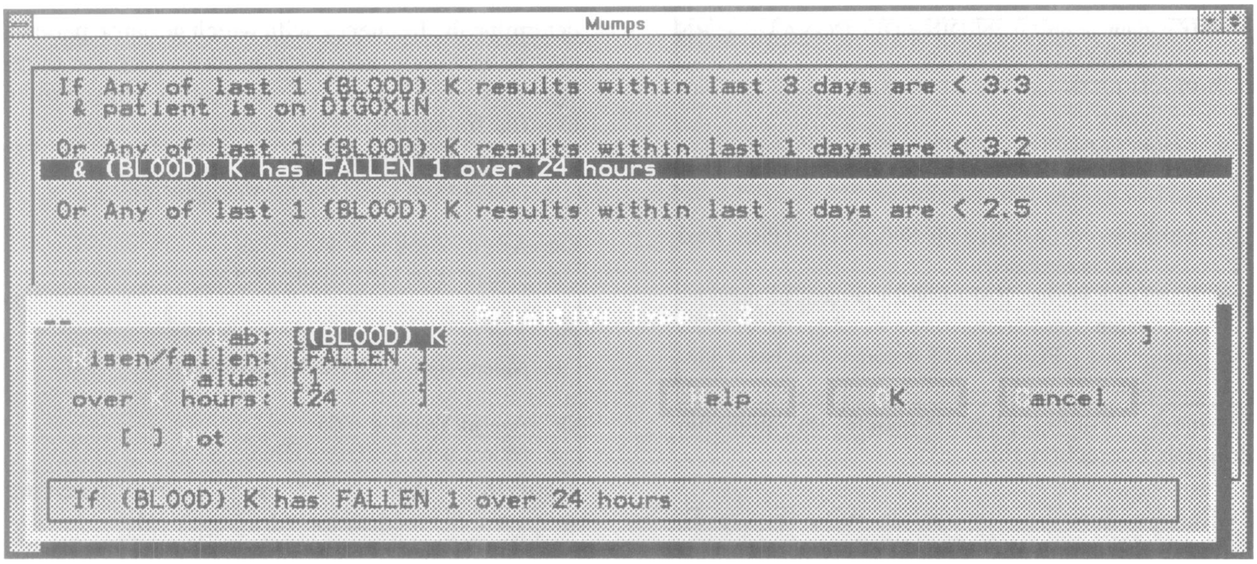

Figure 1. Rule editor screen showing hypokalemia logic being edited. The top part of the screen shows the rule's boolean logic. The bottom part of the screen shows a condition being edited.

VALUE and K would be filled in to instantiate the primitive. LAB would be selected from the dictionary of laboratory tests, RISEN/FALLEN is a binary variable, and VALUE and K are numeric.

Rule editor: We have developed a rule editor in Mumps that has permitted us to write a large number of, and many kinds of, rules by instantiating and combining these primitives. An example of a rule being written, and a primitive being instantiated is shown in Figure 1. The figure shows the logic for a hypokalemia panic lab alerting rule being edited. There are 3 lists of conditions in the rule; the first two lists have two conditions each and the third list has only a single condition. The fields (slots) for the second condition in the second list (K has FALLEN 1 over 24 hours) are being edited.

Categories of rules created: Examples of the different kinds of rules we have created include:

1) Panic lab rules. Seven of Tate's life threatening laboratory alerting rules have been implemented and are being used as part of an automated alerting system. In a recent 4 month period, physicians responded to 342 alerts delivered through an automated link to the paging system. As part of an evaluation of the alerting system, physicians said they would take action as a result of being paged about a panic lab condition in 253 (74%) of the alerts. The logic that defines a falling hematocrit alert for the medical service is shown in Figure 2. Different logic is used for the surgical service (see Discussion).

If HEMATOCRIT has FALLEN 10 since previous result
& Any of last 1 HEMATOCRIT results within last 1 days are < 35

Or HEMATOCRIT has FALLEN 6 since previous result
& HEMATOCRIT has FALLEN .4 per hour since previous result
& Any of last 1 HEMATOCRIT results within last 1 days are < 30

Or HEMATOCRIT has FALLEN 5 since previous result
& Any of last 1 HEMATOCRIT results within last 1 days are < 15

Figure 2. Falling hematocrit panic lab alerting rule.

2) Drug-laboratory interaction rules. Thirty drug-lab interaction rules have been written. Nine are operative as part of the alerting system and the remainder are running in the background to determine the frequency of such events. In a recent 4 month period, 111 drug-lab alerts were conveyed to physicians via the automated alerting system. Physicians said they would "take action" in 58 (52%). In 25 (23%) they said they were already aware of the condition.

The rule that detects a falling platelet count while the patient is receiving quinidine is shown in Figure 3. Representing the concept "quinidine" requires two medication data dictionary elements: quinidine sulfate and quinidine gluconate.

```
If patient is on QUINIDINE GLUCONATE (PRN
      Med Orders excluded)
  & PLATELETS has FALLEN 25 percent of previous
      value
  & Any of last 1 PLATELETS results within last 3 days
      are < 75

Or  patient is on QUINIDINE SULFATE (PRN Med
      Orders excluded)
  & PLATELETS has FALLEN 25 percent of previous
      value
  & Any of last 1 PLATELETS results within last 3 days
      are < 75
```

Figure 3. Falling platelets/patient on quinidine drug-lab interaction rule.

3) Renal failure-drug interaction rules (Figure 4). Strictly, these are also drug-lab interaction rules however they make use of a primitive that infers the presence of renal failure from the patient's sex and age. Thirty renal failure rules have been written and currently are part of the automated alerting scheme. In a recent 4 month period, the GFR rules generated alerts 13 times. In 9 (69%) of those instances the physicians said they would "take action" as a result of the message.

```
If patient does have a GFR<15
  & patient is on PROPOXYPHENE NAP./
      ACETAMINOPHEN

Or  patient does have a GFR<15
  & patient is on PROPOXYPHENE NAPSYLATE
```

Figure 4. Rule to detect patient receiving darvon in presence of renal failure. GFR=glomerular filtration rate.

4) Adverse drug event (ADE) rules (Figure 5). As part of a computerized adverse drug event monitor, fifty automated adverse drug event screening rules have been written. Twenty to forty potential ADEs are written to a file daily for later review by pharmacy personnel and researchers determining the frequency of ADEs at our hospital.

```
If patient is on DIPHENHYDRAMINE HCL (PRN
      Med Orders excluded)
  & previous order for DIPHENHYDRAMINE HCL
      does Not exist over 7 days
  & patient did Not have a Transfusion Order over the
      last 1 days
  & patient is Not on TAXOL (PACLITAXEL)
```

Figure 5. Rule to identify new diphenhydramine order as possible indicator of allergic reaction for automated adverse drug event monitor.

5) Drug-age rules (Figure 6). As part of a study to determine the frequency with which geriatric patients may be receiving inappropriate drugs, twelve drug-age rules were written with the results being written to a file for review.

```
If Age in Years is > 65
  & patient is on CHLORPROMAZINE HCL

Or  Age in Years is > 65
  & patient is on HALOPERIDOL

Or  Age in Years is > 65
  & patient is on THIORIDAZINE HCL

Or  Age in Years is > 65
  & patient is on DROPERIDOL
```

Figure 6. Age > 65 and patient receiving antipsychotic medication (drug-age) rule.

DISCUSSION

We have developed a scheme which represents complex medical events as a boolean combination of simpler events. Further, the simpler events are instantiations of a small number of generic concepts (primitives). Using only a small number of (i.e., 15) instantiable primitive objects, we have represented several different categories of medical events (e.g. panic lab events, drug-lab interactions, etc.) and have started to use these events in a clinical alerting system. The alerting system seems to be moderately well received and and we are studying it further.

We have found this knowledge representation scheme to be quite satisfactory. One attractive feature is the scheme's extensibility. For example, we are currently developing rules that involve a patient's insurance status and admitting diagnosis. All that is required are new primitives "Patient's insurance is INSURANCE" and "Patient's admitting diagnosis is DX" and an extension to the inference engine to evaluate such primitives.

Another attractive feature of our scheme is the ease of maintenance of existing rules. For example, our alerting system originally only applied to the medical service. When we added the surgical service, we changed the limits of the falling hematocrit rule from 35 and 30 (Figure 2) to 28 and we excluded alerts for patients on cardiac surgery altogether. This change took less than a minute and required no programming.

We have chosen the complex conditional approach to knowledge representation over the Arden Syntax, a different approach, for two main reasons. First, the Arden Syntax requires the textual modules to be compiled into native executable code.

The development of such a compiler can consume considerable resources. In contrast, however, the rule editor is a resource not required by the Arden approach. Second, because the approach described herein is declarative, it allows for concepts (e.g., lab value above/below a value) to be reused by simple instantiation. Knowledge base maintenance becomes a relatively simple task of slot filling and database management. Although the Arden Syntax (a procedural approach) allows for meaningful expression of rules and easy readability, two similar rules must still be written as completely separate modules.

One intent of the Arden syntax is to promote the sharing of automated knowledge by providing textual modules that can be shared after minimal manipulation.[13] We wholeheartedly agree that creating sharable knowledge is an important goal in decision support applications development. The textual representation of our rules (Figures 2-6) are not as completely specified as a corresponding Arden module would be but they could well be a suitable starting point for another institution wishing to share our knowledge base. It would be an intriguing experiment to see to what extent our complex conditional rules could be represented using the Arden Syntax, and vice-versa.

Using our rule editor to create rules requires intimate knowledge of the BICS vocabularies (e.g., names of lab tests, medications, etc.) however no programming knowledge is required. Many of the rules listed in this paper were entered by a research assistant with no formal programming experience.

There are some limitations of our restricted boolean form. Most notably is the absence of a distributive property. For example, the concept

$$(\alpha \wedge (\beta \vee \gamma))$$

must be represented as

$$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

(For examples of this issue, see Figures 3, 4 and 6.)

Also, while we have accounted for a few of the most common temporal situations, we have not accounted for any kind of general temporal syntax.[14]

CONCLUSION

We have described a declarative approach to medical knowledge representation that combines simple conditional statements into complex ones. We have used the representation to describe a variety of rules for alerting and data collection purposes at our institution. The scheme is extensible and provides a form of the knowledge that, while not directly compilable, is very readable and could be a reasonable starting point for other groups wishing to share knowledge. The scheme appears to be a good approach to knowledge representation at our institution.

REFERENCES

1. Shortliffe EH. Computer programs to support clinical decision making. JAMA 1987;258:61-66.
2. Musen MA. Dimensions of knowledge sharing and reuse. Comput Biomed Res 1992;25:435-467.
3. Shortliffe EH, chairperson. Report of Medical Informatics Panel: Long Range Plan for National Library of Medicine. National Institutes of Health, United States Department of Health and Human Services, December 1986.
4. Hripcsak G, Luedmann P, Pryor TA, et al. Rationale for the Arden Syntax. Comput Biomed Res 1994;27:291-324.
5. Pryor TA. Design of a knowledge-based hospital information system. SCAMC 1987;11:60-63.
6. McDonald CJ. Action-oriented decisions in ambulatory medicine. Yearbook Medical Publishers. Chicago, IL, 1981.
7. Kuperman GJ, Bates DW, Teich JM, et al. A new knowledge structure for drug-drug interactions. SCAMC 1994;18:836-840.
8. Huff SM, Rocha RA, Bray BE, et al. An event model of medical information representation. J Am Med Informatics Assoc 1995;2:116-134.
9. Musen MA, Eriksson H, Gennari JH, et al. PROTOGE-II: A suite of tools for development of intelligent systems from reusable components. SCAMC 1994;18:1065.
10. Tate KE, Gardner RM, Weaver LK. A computerized laboratory alerting system. MD Comput 1990;7(5):296-301.
11. Rind DM, Safran C, Phillips RS, et al. Effect of computer-based alerts on the treatment and outcome of patients. Arch Intern Med 1994;154:1511-1517.
12. Teich JM, Spurr CD, Flammini S, et al. Response to a trial of physician-based inpatient order entry. SCAMC 1993;17:316-320.
13. Pryor TA. Sharing MLMs: An experiment between Columbia-Presbyterian and LDS Hospital. SCAMC 1993;17:399-402.
14. Shahar Y, Musen MA. RESUME: a temporal absraction system for patient monitoring. Comput Biomed Res 1993;26:255:273.