International Journal of Clinical Monitoring and Computing 8: 43–48, 1991.

Medical Logic Module (MLM) representation of knowledge in a ventilator treatment advisory system

Arkad Kristina,[1] Gill Hans,[1] Ludwigs Ulf,[2] Shahsavar Nosrat,[1] Gao Xiao-ming[1] and Wigertz Ove[1]
[1] *Depart. of Medical Informatics, Linköping University, S-58183 Linköping, Sweden;* [2] *Medical Intensive Care Dept., South Hospital, S-10064 Stockholm, Sweden*

Key words: MLM, Arden Syntax, knowledge representation, knowledge bases, decision support

Summary

In any medical expert system it is the inherent knowledge that is the power of the system and not the particulars of its implementation. Therefore it would be valuable to use a representation that would allow:
- knowledge transfer between different systems,
- users, experts and 'importers' to be able to evaluate the logic,
- experts to easily input their knowledge and be guided how to use the syntax.

Arden Syntax of Medical Logic Module is a proposed knowledge representation, fulfilling these criteria.

The Arden Syntax has been used to represent rules and logic in a decision support system for ventilator therapy in patients with acute respiratory failure (ARF) that is under development.

The medical experts involved in the project have used the Arden Syntax as a convenient way for transfer and storage of medical knowledge. The syntax is easy to learn and may be used with a minimum of training. In the present system, Medical Logical Modules have been used to represent knowledge pertinent to the initiation and maintenance phases of ventilator therapy.

Introduction

Knowledge based systems become increasingly common. The important thing in knowledge based systems is the knowledge and not just the technical manipulation of the knowledge.

Many institutions have developed medical knowledge bases. However, in no single institution is it possible to collect all knowledge in one or more specific domains of knowledge. Therefore the possibility to use knowledge from other knowledge bases would be very attractive, if there existed a way to transfer the knowledge from one institution to another [10]. It should also be easy for a domain expert to read and understand the fact and logic in the knowledge base, as well as to input new fact or modify the old fact. Since new knowledge continuously develops, it is important to maintain knowledge bases and keep them updated. Thus a common syntax for representing knowledge is highly desirable.

The Arden Syntax of Medical Logic Module [1, 2] is a proposed knowledge representation fulfilling these criteria. The Arden Syntax is developed jointly between groups at the University of Utah, Salt Lake City, Columbia University, New York, Erasmus University, Rotterdam and Linköping University. We have used the Arden Syntax to represent rules and logic in a decision support system for ventilator therapy in patients with acute respiratory failure (ARF) that is under development [3, 9]. This is a project in cooperation with South Hospital in Stockholm, Siemens-Elema, Solna and Linköping University. The project is called

```
maintenance:
  title:           Weaning criteria,vitals;
  filename:        \wp\kusmlm\.mlm;
  version:         1.00;
  institution:     Miva,SöS;
  author:          Ulf Ludwigs;
  specialist:      ;
  date:            '1990-03-10'; '1990-07-05';
  validation:      testing;
  format:          ISO;

library:
  purpose:         Checks for vitals indicative of weaning success;
  keywords:        wean;
  citations:       1.   Sahn SA, Lakshminarayan S, Petty TL.
                        Weaning from mechanical ventilation.
                        JAMA 1976; 235:2208-2212
                   2.   Hall JB, Wood LDH. Liberation of patient
                        from mechanical ventilation. JAMA 1987;
                        257:1621-1628;
  links:           ;

knowledge:
  type:            datadriven;
  data             gcs := READ LAST {gcs};
                   hr := READ LAST {hr};
                   temp := READ LAST {temp};
                   map := READ LAST {map};
  priority:        ;
  evoke:           ;
  logic:           IF gcs>10 and hr<100 and temp<38 and map <90
                   THEN message := "Vital parameters and level of
                   consciousness adequate for a weaning trial"
                   ELSE message := "Vital parameters and/or level of
                   consciousness presently contraindicates a weaning
                   trial";
                   Conclude(true);
  action:          STORE message TO {user};
```

Fig. 1. An MLM which checks for vitals indicative of weaning success. The variables, gcs = consciousness, hr = heart rate, temp = temperature, map = mean blood pressure, are retrieved from the local database.

KUSIVAR (Knowledge-based User Support In Volume-controlled Artificial Respiration). It has been going on since 1986 but Medical Logical Modules were introduced at the end of 1989.

Physicians have described and written rules and logics with help of the Arden Syntax. So far, the entry of Medical Logic Modules is supported by wordprocessing macros and the translation to the internal KUSIVAR-format is done manually.

The MLM concept

The Arden Syntax of Medical Logic Modules, The Arden Syntax for short, is focused on knowledge that can be represented as a set of independent modules, Medical Logic Modules (MLM's). The idea behind this is that a module can be understood and used without referring to the rest of the knowledge base. This is just an approximation because most real systems do not satisfy the independent requirement perfectly due to nesting and special properties. It is possible to incorporate the MLM's one by one in a local system, as they are independent. An institution might need to check the references or adjust probabilities to fit into the local system. The independency makes it possible for an institution to test the MLM's before they are incorporated into the local decision support system. This makes it possible to incorparate a whole decision support system or parts of it into a local system.

The syntax of MLM is based on Pascal and SQL, and has derived many properties from the HELP-[8] and CARE-[5]systems.

An MLM consists of slots, grouped into categories. Every slot starts with a slot name which identifies the slot, continues by a slot body which contains the data and ends with a semi-colon.

The first category is the maintenance category which contains those slots that specify information unrelated to the medical knowledge in the module. The title slot describes in short what the MLM does, it serves as a comment. All titles and filenames must be unique. The version slot should be changed when the MLM's are updated. The specialist slot tells which person in the institution who is responsible for validating and installing the module. This slot should always be present but blank when transferring MLM's from one institution to another. The receiving institution accept responsibility for use of the module by fill this slot. The format slot is not required. However, it specifies the format of the MLM, at present only the representation of dates and times. ISO is the default format for this slot.

The library category contains those slots that are related to the module's knowledge except the slots which belong to the knowledge category. The purpose slot explains the purpose of the module, and serves as a comment. The keywords slot contains descriptive words used for searching (like MeSH-terms for Medline).

The knowledge category contains those slots that actually specify what the module does. The type slot is included to allow future expansion (there is only one type so far). The data slot defines access to the patient database. The evoke slot says when the decision module becomes active. The logic slot contains the actual medical rule or medical condition to test for. The action slot specifies what to do if the logic slot becomes true.

There are a few basic goals which are critical to make the MLM standards useful for sharing knowledge.

Readability
The MLM's should be able to be read and interpreted easily by medical experts with little training.

Lack of ambiguity
The MLM's should be unambiguous so that the same module cannot be interpreted in two different ways.

Modules can be ambiguous in two different ways, at first, the parser of the computer may interpret a module in different ways and, secondly, the user can interpret a module in different ways.

Writeability
The MLM's should be able to be written by medical experts with as little training as possible. Note that writeability and readability differ from each other. For example, it is easier to recognize a list of numbers like '1, 2, 3;' than it is to remember that the list is delimited by commas and terminated with a semi-colon. It does not matter what syntax one use, it will make it more difficult to write the MLM's if one wants to make it strictly unambiguous. One can produce a special editor to facilitate writing the MLM's, but the modules must be readable in their original form.

Ease of computer translation
The MLM's should allow easy computer translation. This is less important than the first three requirements.

Ease of maintenance
It should be possible to manipulate and maintain the MLM's on common computer environment.

In clinical medicine time is important. There need to be powerful yet concise operators to specify time intervals. In the Arden syntax time intervals are defined in terms of anchor points, which are absolute points in time, duration, or time intervals.

Expression like
- 3 months ago
- 2 days after time of surgery
- 2 months before now

are supported.

All queries to the patient database return multiple instances unless an aggregation operation says otherwise. So a query for a patient's hemoglobin level during the last three days may return a set of answers that fulfil the time constraints. Only by using aggregation operations like last or minimum a single element can be returned.

The KUSIVAR-project

The main objective of the KUSIVAR-project is to develop a decision support system for ventilator therapy in patients with acute respiratory failure. The system is being designed to provide:
1. advanced ventilatory monitoring including efficient data interpretation and presentation.
2. expert advice concerning ventilator strategy based on information from the monitoring system.

It will also relieve the staff from tasks which are well suited for automatic control, like optimization of ventilator settings and adjustment of ventilatory volumes.

The project has been going on since 1986 and a prototype has been built on an Explorer workstation from UNISYS [9]. This machine can not work as a data-driven machine in that sense we want and it is also too expensive to be used as a bedside monitor. A PC-based system is under development. This system is data-driven and uses forward chaining in the way that the system can generate and display advices automatically if there are any changes in data from the ventilator or the patient.

The main blocks in the KUSIVAR-system are the knowledge-base, the computer, the ventilator and several databases. The computer is a Compaq 386 and the ventilator is a Siemens-Elema, Servo-Ventilator 900 C. To be able to collect data from

```
maintenance:
  title:          Set initial frequency;
  filename:       \wp\kusmlm\rrini.mlm;
  version:        1.00;
  institution:    Miva, SÖS;
  author:         Ulf Ludwigs;
  specialist:     ;
  date:           '1990-01-17';'1990-07-05';
  validation:     testing;
  format:         ISO;

library:
  purpose:        Determines initial ventilator-setting of RR.;
  keywords:       Initiation, RR;

knowledge:
  type:           datadriven;
  data:           dia:= READ LAST {man.diag}
                  rr:= READ LAST {man.manrr};
  priority:       ;
  evoke:          ;
  logic:          IF dia = "BA" OR dia = "COPD" THEN
                     IF rr > 15 OR rr <35 THEN
                          resp_rate := rr
                     ELSE resp_rate :=30
                  ELSE resp_rate := 20
                  CONCLUDE(TRUE);
  Action:         RETURN resp_rate;        /*To messini.mlm*/
```

Fig. 2a. An MLM which determines initial ventilator-setting of respiratory rate. The variables, dia = diagnose, rr = resp rate, are retrieved from the local database.

```
RULE:     Rule 19
If        RR.lastfive is greater than 15
          And RR.lastfive is less than 35
Then H1_RRINI
          is confirmed.

RULE      Rule 21
If        DIAG.last_s is "BA","COPD"
          And there is no evidence of H1_RRINI
Then H2_RRINI
          is confirmed.
          And 30 is assigned to RESP_RATE

RULE:     Rule 22
If        DIAG.last_s is "BA","COPD"
          And there is evidence of H1_RRINI
Then H2_RRINI
          is confirmed
          And RR.lastfive is assigned to RESP_RATE
```

Fig. 2b. This is rules which are Nexpert rules based on the MLM in Fig. 2a. The variables, DIAG = Diagnose, RR = Resp rate, are retrieved from the local database and H1_RRINI and H_2–RRINI are local Nexpert variables.

the 900C a Servo Computer Module (SCM) is used. It is possible to enter values manually and to use values from ventilator monitoring devices. In the future it will be possible to use data from the intensive care units general database. The software is implemented as a number of modules under the Microsoft Windows multitasking environment. Among the software modules are programs for communication with the Servo Computer Module, the database handler and programs for manual data entry. All values are stored in dBase-files. The knowledge base is implemented with help of Nexpert Object [6].

The system should be a real-time system, because if the decision support system for ventilator therapy does not get the data from the ventilator fast, or can not deliver the advice fast, the decision support system will be of limited use. The hospital staff is in control of and of course responsible for the therapy. The decision support system will act as a help for the staff, by helping them to collect data, put the data together and give an advice if it is requested.

The ventilator treatment contains three phases:

1. Initiation, where indications or contraindications for mechanical ventilation are evaluated.
2. Maintenance, where an optimization of the ventilator settings is done using an advanced monitoring of the respiratory function of the patient as well as an automatic detection and recognition of complications.
3. Weaning, where strategies for termination of the ventilator treatment is planned.

In the maintenance phase the decision support system helps to determine the optimal settings of the ventilator. The approach is to use advanced mathematical optimization methods based on on-line values from equipment around the patient combined with rule-based logic [3]. This optimization can hopefully shorten the time for the patient in the ventilator. The final goal of the optimization is to reduce morbidity and eventually mortality by frequent changes of ventilator settings in response to the dynamic variations in pulmonary functions.

Result

MLM's have been used, in the KUSIVAR-system, to represent knowledge pertinent to the initiation and maintenance phases of ventilator treatment. The medical experts involved in the project have turned to use MLM as a convenient way for transfer and storage of medical knowledge. It provides flexibility, but knowledge entry is slow. The syntax is easy to learn and may be used with a minimum of training.

The main problem we had, and still have, is in the translation from MLM-format into NEXPERT knowledgebase. One problem is that the construction 'IF <expr> THEN <expr> ELSE <expr>' is missing. That gives the result that one MLM, connected via a net of subgoals, gives many NEXPERT-rules (Fig. 2a and b). This gives an inefficient code which makes the decision support system slower. And the code in the knowledgebase gets harder to read and understand. As we translate all rules manually we have found that an compiler or interpreter which can do this job should be very useful.

Discussion

It is obvious that the Arden Syntax is useful in describing medical knowledge. But if the specialist does not want to use a lot of time in trying to insert the module into the knowledge base it is important that he has available some kind of editor which can check the syntax and have connections with a medical entity dictionary. A compiler is needed to transform the MLM into executable format.

There may be two different ways of realizing decision support logic. One is to compile each logic module into a 'stand-alone' program which is called by the controller [7]. The other approach is to precompile each module into pseudo-code which is executed at run-time by a special pseudo-code interpreter. Since a lot of decisions have to be made dynamically at run-time and we also require knowledge base extension to be a fast process the pseudo-code alternative may be preferable. In any case a number of tables and inverted lists (e.g. for module cross-linking) have to be built during compilation to improve execution speed.

The main functions of the pre-compiler are to

- perform syntax- and lexical analysis
- check the variables with the medical dictionary and perform type checking
- transform the evoking criterias into elementary functions
- transform the complex MLM expressions into elementary request lists
- connect the necessary retrieving processes to the variables
- maintain tables with inverted lists of used variables, functions and operations, MLMs, evoking criterias and different kinds of cross references.

After introducing the Arden Syntax to the physicians, we found that they needed a powerful aid for the writing. This aid can be some kind of an editor with syntax control. It is hard to remember all details in the module format. The syntax of the modules must be correct if the MLM's should be accepted by the computer. Waiting for a complete editor, one of the domain experts created his own tool for rule entry. A WordPerfect Macro gives the keywords he needs and makes it easy to follow the module layout.

However the main functions of an MLM editor should be:

- a syntax directed creation of new MLM modules
- to provide standard format MLM-modules
- on-line help to retrieve and incorporate medical terms from a medical entity dictionary, which includes all the terms that are used in the application area and arranged in a logical and work environment oriented way
- to handle lists of MLM-modules in the knowledge-base and update different kinds of cross-reference tables
- to provide add-, delete-, replace-, copy- and edit functions
- to check the knowledge-base for redundancy and contradictions

Most of these functions are non-trivial and contains general research problems, like the use of graphical interfaces for knowledge acquisition, providing tools for syntax checking, links and mapping facilities to standardized vocabularies, user-friendly functions for maintenance of the knowledge-base and probably the biggest problem with huge knowledge-bases in the future, the issues of integrity, contradictions and redundancy.

If different software components like data bases and knowledge bases should be able to communicate, they are supposed to be able to use the same data. To facilitate this data sharing a common medical entity dictionary is needed. The knowledge-base have to be connected to a patient database so that only questions regarding data that are not already in the data base should be asked for.

A proposed system design for the realization of Medical Logic Modules as working modules in a patient information system has been reported [4], which discusses different ways to solve many of these issues.

Conclusion

The Arden Syntax of Medical Logic Modules seems to be a valuable help to describing knowledge.

The Arden Syntax has the following features:

- it is a proposed standard. This gives the opportunity to share knowledge between different users.
- it helps domain experts to be less dependent of knowledge engineers, in describing their knowledge.
- it is easy to write with minimal training. However it is difficult to remember all details.
- it is easy to read by medical experts with only little computer training.
- it is unambiguous.
- it facilitates maintenance of the knowledge base.

The medical experts involved think that the Arden Syntax give them flexibility to formulate the rules in a way that seems natural to them.

References

1. Arden Homestead Medical Logic Module Standard Ver. 1.11, June 12, 1990.
2. Clayton PD, Pryor TA, Wigertz OB, Hripcsak G. Issues and Structures for Sharing Medical Knowledge among Decision-Making Systems: The 1989 Arden Homestead Retreat. In: Kingsland LC (ed) Proc 13th Ann SCAMC 1989. IEEE Comp Soc Press, Washington DC, 1989, pp 116–21.
3. Gill H, Ludwigs U, Matell G, Rudowski R, Ström C, Wigertz O. Integrating knowledge-based technology into computer aided ventilator systems. Int J Clin Monitoring & Computing 7: 1–6, 1990.
4. Magyar G, Arkad K, Ericsson K-E, Gill H, Linnarsson R, Wigertz O. Realizing medical knowledge in MLM form as working modules in a patient information system. Proc IMIA Working Conf on Software Engineering in Medical Informatics. Amsterdam 1990.
5. McDonald CJ. Computer-Stored Medical Record Systems. M.D. Computing 5: 4–5, 1988.
6. Neuron Data Inc: Nexpert Object Version 2.0 User's guide.
7. Pryor A, Dupont R, Clay J. A MLM based order entry system: the use of knowledge in a traditional HIS application. SCAMC 1990, pp 579–83.
8. Pryor TA, Gardner RM, Clayton PD, Warner HR. The HELP System. J Med Syst 7: 87–102, 1983.
9. Shahsavar N, Frostell C, Gill H, Ludwigs U, Matell G, Wigertz O. Knowledge base design for decision support in respirator therapy. Int J Clin Monitoring & Computing 6: 223–31, 1989.
10. Wigertz OB, Clayton PD, Hripcsak G, Linnarsson R. Knowledge Representation and Data Model to support Medical Knowledge Base Transportability. Meth Inform Med 1989. In press.

Address for offprints:
K. Arkad,
Dept. of Medical Informatics,
Linköping University,
S-581 83 Linköping, Sweden