



0010-4825(94)00027-1

## THE USE OF MEDICAL LOGIC MODULES AT LDS HOSPITAL

T. ALLAN PRYOR

Intermountain Health Care, 36 S. State Street, Suite 800, Salt Lake City, UT 84111, U.S.A.

(Received 3 June 1994; received for publication 21 July 1994)

**Abstract**—The development of medical knowledge bases for use in a clinical information system (HELP) has been an ongoing goal at LDS Hospital in Salt Lake City, Utah, for the past 25 years. In building our medical knowledge base we felt the need to implement a decision support syntax which could capture the logic of our experts in a way that was not only executable, but also easily read and shared by others. During these 25 years we defined several simple syntaxes to express this medical logic. Our current approach is to cooperate with international standards groups (ASTM) and use the Arden Syntax for medical logic modules. We are working with the 3M Corporation in the joint development of an Arden Compiler for HELP. We plan to use the Arden Syntax initially to support our alert/reminder system and computerized management protocols.

Clinical decision making    Alerting systems    Hospital information systems

### 1. INTRODUCTION

For over 25 years at LDS Hospital in Salt Lake City, Utah, we have been developing a large knowledge-based hospital information system (HIS) which we refer to as HELP (health evaluation through logical processing) [1, 4, 5]. A primary goal for the HELP system at LDS Hospital has been the development of a HIS which not only facilitates the capture and display of clinical and financial data, but is also able to assist the caregiver in management of patients through computer-generated medical decision suggestions. These suggestions are generated through the evaluation of medical logic stored in the HELP knowledge base. To create this knowledge base we have been working with physician and nursing experts over a 20 year period. To facilitate this interaction with our experts we wanted the system to be sufficiently intuitive that they could independently create and maintain the medical logic for which they were responsible without the assistance of dedicated programmers. To achieve that goal we felt it necessary to define a medical logic syntax that was easily understood by our experts. The creation of such a syntax has been an evolving process in the 25 year history of HELP. The culmination of this syntax is our adoption of the standard Arden Syntax for medical logic modules (MLMs). The following is a brief outline of ongoing efforts over the years to arrive finally at our use of Arden Syntax.

Our first attempt at definition of a medical logic syntax was called HCOM for HELP Compiler. This syntax was defined in the mid-1970s. It allowed the expert to write a logic module which consisted of either one or more Boolean statements or the sequential evaluation of the Bayes formula with independent variables. Each logic module was referred to as an HCOM "sector" since the amount of logic possible in a single sector could not exceed the physical size of the Control Data Corporation disk drives used at the time. The HCOM sectors were compiled to a set of pseudo-codes which were then interpreted at time of execution of the sector. The physical size limitation constrained the compiled module to less than 1024 bytes of memory. While this physical limitation was severe, we were able to write hundreds of sectors which provided interpretative or alerting logic in several domains including ECG interpretation, blood gas interpretation, and clinical laboratory alerting.

In 1980 we replaced our Control Data Corporation computers with Tandem computers. With this conversion we felt it necessary to enhance the capability of the HCOM syntax. This enhancement became known as the HELP language [2]. The HELP language provides additional syntax which allowed us to develop MLMs of increased complexity. At the same time we began development of a HELP-specific database language to increase our productivity in writing application programs using the HELP database system, referred to as PTXT. This language we called PAL for PTXT application language. It soon became the standard language for creation of most of the applications resident on HELP. PAL was designed to provide a relational view of the patient data stored in our proprietary PTXT database. PAL was designed to incorporate many of the features now known as object-oriented programming. Because of the features of the Tandem operating system Guardian, we again decided to compile the PAL syntax to a set of pseudo-codes which would be interpreted at run time. While this meant that the execution of the PAL syntax would be slower, because of some of the low level HELP library routines, execution of the database retrieval features of PAL resulted in an overall adequate performance for the system. With the development of PAL as a general application programming language we decided that the HELP language compiler should then compile the HELP syntax into PAL syntax which would then be compiled into pseudo-codes for execution on the system. Because of the limited nature of the HELP language we also provided syntax within HELP to imbed PAL statements directly in the HELP MLMs.

Because of our desire to share the medical knowledge which we and others were developing, we enthusiastically participated in the original Arden Conference and supported the decision to develop a standard syntax to be used internationally. Having made that decision and with the rapid development of the Arden Syntax we have discontinued our development of the HELP language and have begun moving to the standard Arden Syntax as the standard syntax for writing MLMs on the HELP system. Because of our partnership with the 3M Corporation (3M currently has the rights to HELP and is continuing to develop and market the HELP products) for the joint development of HELP, the development of the Arden compiler for HELP is actually being done by 3M. Their initial efforts in the development of the HELP Arden compiler closely parallels the implementation of the syntax used for the HELP language. That is, the Arden compiler produces PAL syntax which is then compiled for execution on HELP.

## 2. ARCHITECTURE FOR IMPLEMENTATION OF ARDEN AT LDS

The implementation of Arden on any system requires the site to define the syntax which they will use to query their local database. That site-specific database query syntax is placed within the curly brackets of the Arden **Data** slots. The database query syntax used at LDS is a carry-over from our early HELP syntax. The following defines the basic syntax used for defining HELP data elements using the HELP syntax:

Label **WHICH IS** <PTXT Code >

The Label refers to variable name to be used in the MLM and would generally be the same as the variable label used outside the curly brackets in the definition of the logic variable. The PTXT Code is an 8 byte numeric defining the code used within HELP to identify a data element. The following example illustrates the Arden Syntax which would be used at LDS to define in Arden a variable **digoxin** which is the latest digoxin level measured on a patient during the last 3 months.

**digoxin** : = **READ LAST** {**digoxin WHICH IS** 17.1.8.1.1} where it occurred within the past 3 months;

In the above example the code 17.1.8.1.1 is the PTXT code in the HELP data dictionary to define the code used to store the value of the digoxin level. Multiple data variables are defined in a single Arden clause by delimiting their definitions with commas. While the **WHICH IS** syntax is the most common, more complex data queries

are possible. These additions to the simple variable definition are used to direct the Arden compiler to incorporate some of the special features of the HELP system. Among those features is the ability to designate the source of the database to be queried. For example, if the data to be used were the incoming data being stored in the patient's database (in common with many data-driven applications), the extension **SOURCE** buffer would cause the compiler to create a code which only searched the global memory space available at the time of execution of the MLM and would not perform a database query from the disk repository. The above example would be modified to the following if such were the requirements of the query.

digoxin := **READ LAST** {digoxin **WHICH IS** 17.1.8.1.1 **SOURCE** buffer} where it occurred within the past 3 months;

This construct has proven to be quite useful in HELP where in many instances the call for execution of an MLM has resulted from some previous query from the HELP database or is a result of the data-driven triggering of the MLM. Other similar features have been added to HELP to maximize the performance of the system.

The second site-specific concern is the transmission of the alerts to the appropriate destination within the system. For the LDS implementation we have chosen to write the alerts both to the patient's record and a special alert file. Within the alert file, the alert is also tagged to identify an individual or specific location which should be notified when this alert occurs. In writing an alert for the HELP system we include not only a message indicating the an alert has occurred, but also the actual critical patient data which were used to trigger the alert. By including this critical patient data it is possible to easily provide the caregiver pertinent clinical data about the patient at the time they acknowledge the alert. Our alert file is a circular file which has been sized to accept about 1 week's set of alerts. Writing into this file automatically triggers other features of HELP which are used to notify the appropriate caregivers of the occurrence of an alert. It uses tables within HELP to route the alert to the correct destinations, and when the alert is acknowledged by the user it will link the acknowledgement screens to any special screens which are associated with the particular alert. For example, if the caregiver chooses not to acknowledge an alert or complete any actions suggested by the alert, alert-specific screens are presented to the user which attempt to capture the reasons why the alert is being rejected. These features of our alerting system are used to provide the continuous feedback necessary to understand the acceptability of our MLMs and, when needed, to improve the logic of our MLMs.

### 3. EXAMPLES OF DECISION SUPPORT IMPLEMENTATION AT LDS

The use of decision support as mentioned above has been one of the primary benefits of the HELP system at LDS [3]. While 3M has taken on the task of writing the HELP compiler, this development is not yet complete; thus the logic modules which are currently running clinically have not yet been rewritten in Arden Syntax. The examples described in this section will, however, illustrate the uses of decision support and serve as models for 3M to ensure that the HELP Arden compiler will support the functions necessary to implement the decision applications currently available on the HELP system at the LDS Hospital. The examples reported here have all been written in the HELP/PAL language and exist as independent modules on HELP.

There are two main areas for which most of our MLMs are being written. They are the continued creation of alerts/reminders and the implementation of computerized protocols. The alerting modules are generally data driven. That is, they are activated at the time of storage of a data element into the patient database. Examples of such alerts include the Clinical Laboratory alerts which monitor the laboratory values as they are placed in the patient's record. Other alerting modules include our medication critiquing modules which critique all medication orders as they are placed on the system. Notification of the alerts has traditionally been through either terminal messages or printed reports. Our latest efforts, however, have been to incorporate the alerting system

with a nursing electronic beeper system. With this link the alert is automatically transmitted to the "beeper" system and the nurse responsible for care of the patient is notified through the beeper she/he carries. If no acknowledgement of the alert has been completed through a HELP terminal within 15 min, a second beeper notification is sent to the nurse. These notifications will continue until either the alert has been acknowledged or 2 hr has passed since the time of the alert. Our initial experience with this link has been very positive, with good acceptance by our nurses.

Our second area of decision support focus is the creation of computerized patient management protocols. Our early efforts in computerized protocols were in the management of patients with adult respiratory distress syndrome. Because of the success of this early work we are continuing to expand the number of protocols available on the system. Some of the newer protocols which we have recently installed are (1) management of heparin therapy, (2) management of pressure ulcer sores, and (3) nursing protocols for the management of blood pressure and fluid balance in our Thoracic Intensive Care Unit (ICU). The implementation of these protocols has used different execution strategies. The heparin protocol is a data-driven protocol. That is, the protocol is automatically executed when selected data elements are stored in the patient's database. Notification of the protocol suggestions is routed directly to the nurse in charge of the patient through our link to the nursing beeper system. The Thoracic ICU nursing protocols are interactive and are executed on demand by the nurse. That is, the nurse has an option on her/his nursing documentation screen which if chosen will execute the nursing protocol and provide care suggestions to the nurse. Finally, the pressure ulcer protocol has been implemented as a natural part of our computerized nursing documentation application and must be noted when ulcer care is given to the patient. It is the diversity of both protocols and the mechanisms of implementation of these protocols that are continuing to grow at our site.

#### 4. FUTURE DIRECTIONS AT LDS

The future of Arden at LDS is closely tied to the future of the HELP system at LDS. Currently, we are beginning expansion of the HELP system to include networking to our out-patient clinics and physician offices. This expansion will result in a network-wide longitudinal patient data repository (LPR). HELP applications which will be placed in the physician's offices will utilize the enterprise LPR as their primary store. Applications will be written for a GUI client front end with the enterprise LPR becoming the server to those clients. We plan to include within this system the same decision support which is now available on the Tandem HELP system. 3M will develop the Arden Compiler for us. Similar data-drive and time-drive triggering events will also be supported. Applications on the system will be written using an object-oriented visual programming language. The output of the Arden Compiler will be executable objects. We will be able to execute these objects as in the current system through direct calls from the application programs or through either the data-driven or time-driven triggers. We plan to have this system available for initial clinical testing within 1 year. Following that period we will replace many of the Tandem-based in-patient applications with this new paradigm.

We also plan to work with our Home Health Care nurses to provide them with portable computers which will run some of the newer applications and assist them in the management of patients in the home. We will include within their applications management protocols which we plan to develop using the Arden Syntax. Since those applications will be linked to our central data repository, alerts may be directed not only back to the nurses, but also to the physicians who have responsibility for the patients. This notification will take advantage of the enterprise network we are building.

#### 5. SUMMARY

This paper describes the ongoing work at LDS Hospital in Salt Lake City, Utah, in the development of a medical logic syntax for creation of decision support modules for the

HELP system. Our early work in this area resulted in several languages which were to some extent the forerunners of the Arden Syntax. Using these languages we have developed numerous (over 1000) MLMs. These modules are used on HELP to alert users to abnormal patient conditions, critique orders, interpret medical data patterns, and implement computerized patient management protocols.

We are working with the 3M Corporation, which has the rights to HELP, in the development of an Arden Compiler for HELP. When this compiler becomes fully operational to us, we plan to convert our existing logic modules to the Arden Syntax in the hope of sharing that logic not only with other HELP users, but with the medical community at large. In our compiler development we are using as our site-specific syntax the earlier syntax which we have used in our HELP language efforts.

Within Intermountain Health Care the parent company of LDS Hospital we are now installing the Help system throughout their hospitals in Utah. With this network of hospitals we will be building an enterprise-wide patient data repository which will be extended to support the hospital data on our patients, and will be linked to a HELP out-patient system which will store information from the out-patient clinics, physician offices, and home-care visits. We will provide similar decision support capability for this data repository to that now provided for the hospital setting. This will require us to develop another version of our Arden compiler which runs on an open architecture environment and has access to this complete data repository.

#### REFERENCES

1. T. A. Pryor, J. D. Morgan, S. J. Clark, W. A. Miller and H. R. Warner, HELP: a computerized system for medical decision making, *Computer* (Jan) 34-38 (1974).
2. M. R. Ostler, J. D. Stansfield and T. A. Pryor, A new efficient version of HELP, *Proceedings of the Ninth Annual Symposium on Computer Applications in Medical Care*, pp. 296-297, November. Computer Society Press, Washington, DC (1985).
3. T. A. Pryor, R. D. Clayton, P. J. Haug and O. Wigertz, Design of a knowledge driven HIS, *Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care*, pp. 60-63. Computer Society Press, Washington, DC (1987).
4. T. A. Pryor, H. R. Warner, R. M. Gardner, P. D. Clayton and P. J. Haug, The HELP System Development Tools, *Implementing Health Care Information Systems*, pp. 365-383 (1989).
5. T. A. Pryor, Development of decision support systems, *Int. J. clin. Monitoring Computing* 7, 137-146 (1990).

**About the Author**—T. ALLAN PRYOR received his Ph.D. in Computer Science from the University of Utah in 1972. He is a fellow of the American College of Medical Informatics. His research interests for over 30 years has been the development of hospital-based information systems to support the clinical care of patients.