*International Journal of Clinical Monitoring and Computing* **10**: 215–224, 1993.


# The arden syntax for medical logic modules

T. Allan Pryor[1] & George Hripcsak[2]
[1] *Department of Medical Informatics, LDS Hospital/University of Utah, Salt Lake City, UT 84143;* [2] *Center for Medical Informatics, Columbia-Presbyterian Medical Center, New York, NY 10032, USA*

Accepted 22 March 1993

## Abstract

The Arden Syntax for sharing medical knowledge bases is described. Its current focus is on knowledge that is represented as a set of independent modules that can provide therapeutic suggestions, alerts, diagnosis scores, etc. The syntax is based largely upon HELP and the Regenstrief Medical Record System. Each module, called a Medical Logic Module or MLM, is made of slots grouped into maintenance, library, and knowledge categories. The syntax has provisions for querying a clinical database and representing time. Several clinical information systems were analyzed and appear to be compatible with the syntax. The syntax has been tested for syntactic ambiguities using the tools lex and yacc. Seventeen institutions are currently in the process of adopting the Arden Syntax for their decision-support systems. A subcommittee of ASTM has been formed to develop standards for sharing medical knowledge bases. The Arden Syntax has been published by ASTM as a initial standard for sharing medical knowledge.

## Introduction

From its beginning Medical Informatics has focused on the use of the computer as an adjunct to the decision making of the health care professional. Creation of expert systems has been a major emphasis of many researchers in Medical Informatics. Early examples of expert systems includes: Dr. Bleich's pioneering work on the interpretation of acid base disorders [1], Dr. Warner's development of the HELP [2, 3] system for generalized medical decision support in the hospital, Dr. McDonald's use of reminders in the outpatient setting (CARE) [4–6], Dr. Meyers' and Dr. Miller's creation of an expert system for diagnosis of internal medicine diseases (Internist) [7], Dr. Shurtliffe's pioneering work on rule based expert systems for recognizing and treating bacterial agents (MYCIN) [8], and Dr. Barnett's diagnostic decision-support system DXplain [9]. Each of these efforts were expanded by investigators at those institutions and new research group soon emerged at other institutions with new ideas for computerized clinical decision support. Like all new technology, this exploding research was uncoordinated and left to the individual research group to investigate mechanisms for description of the medical logic and techniques for delivery of the logic to the health care providers. By the mid-1980s it was apparent that, used properly, computerized delivery of medical logic was effective in enhancing the quality of care provided to patients.

With this success, however, it became clear that no single center could produce the entire knowledge base necessary to cover all of the decision support useful in all facets of clinical medicine. Eventhough multiple groups were having success in limited areas, the one disrupting fact was that as each institution engaged in research, the investigators has chosen their own methodology of their decision logic. The ability to share that logic among institutions was almost nonexistent. While numerous papers were presented outlining medical logic validated at one site, duplicating that logic on systems at other institutions was difficult and required almost as much effort to validate and implement as that done by the original site. It was felt that this effort could be greatly reduced if there were standards for sharing individual knowledge bases among research centers. With standards for description of expert technology, the work of each of the research centers could be combined to create a large clinical knowledge base available to all.

### Sharing medical knowledge

To address this issue, a workshop was held at Columbia's Arden Homestead conference center on June 16–18, 1989 [10–13]. This conference brought together many of the leaders in Medical Informatics throughout the United States and Europe. The specific goal of the conference was to measure the interest throughout the Informatics community in sharing medical logic.

The first task of the group was to determine if any medical logic could be shared. After review of many of the existing artificial intelligence and expert systems reported in the literature, it was decided that only a subset of the knowledge structures reported were currently candidates for standardization. The most prominent among these was a class of decision logic described as modular independent logic. This class, exemplified by much of the knowledge base of the HELP and CARE systems, was one wherein a single rule could exist and be of value, independent of a large integrated knowledge base. An example of such an independent rule might be one to alert the physician ordering aminoglycoside that it is contraindicated because the patient has laboratory values indicating poor renal function. In this example, the rule to alert the care giver of the potential adverse effect of the drug is independent of any other knowledge which might exist and could be utilized in a system even if it were the only piece of knowledge known to the system.

A second class of logic, modular interdependent, was also described. This class, while still modular, required a set of interdependent modules or rules to be fired in order for the correct decision to be determined. Many of the diagnostic systems like DXplain [9], Iliad [14], and QMR [7] fit well into this category.

The last class of medical decision logic considered consisted of those highly interdependent artificial intelligence systems involving semantic and neural nets, etc. This class of logic was felt to be beyond the ability of the conference to address.

The unanimous opinion among the attendees was that the first class of logic, modular independent, was well suited for standards considerations. The name given to these modular independent rules was Medical Logic Modules (MLMs). (An example of an MLM is shown in Fig. 1 and is described later in this article.) With this decision made, the conference concentrated on the difficult task of trying to describe a standard syntax that could be used to define the logic contained in the MLM. In developing a standard syntax, the goal was to develop a common language to describe this restricted class of medical rules/decisions without regard to the actual implementation of that logic in existing systems. It would be the responsibility of each recipient of the MLMs to determine how the logic described in the MLM would be compiled for implementation on their medical decision support system. By concentrating only on the syntax for describing the knowledge we felt that the issues of proprietary systems would be avoided. That is, the knowledge would be availabe to all, but the system on which the logic was executed would in many cases be proprietary to a particular vendor. By skirting the proprietary issue it was hoped that more groups would be encouraged to participate in the sharing of medical knowledge bases.

### Syntax alternatives

Given this decision, the next step was to decide whether to adopt an existing syntax, or to define a new syntax for sharing knowledge. There were several alternatives. General-purpose programming languages are certainly expressive enough to define medical logic. Their main problem is that they lack medically relevant constructs and data structures such as database queries and time. If each institution is free to implement these items as it pleases (e.g., its own database implementation of time), then sharing between institutions could require significant translations.

General purpose query languages like SQL can retrieve data but do not contain the constructs needed to compare data in complex ways. For example, SQL lacks the block structure that would be provided by an IF-THEN statement, and it lacks constructs for time.

Other alternatives include choosing the syntax's of existing decision-support systems. The syntax of Health Evaluation through Logical Processing (HELP) [16] and the syntax of the Regenstrief Medical Record System (RMRS) [4–6], are of particular interest because both support modular independent knowledge, both have demonstrated clinical success, and both are well established in their sponsoring institutions. One problem is that each of these systems supports queries only to its own proprietary patient database. The queries have not been designed specifically for the purpose of sharing knowledge. Furthermore, some systems (like HELP) contain constructs for purposes other than for making medical decisions. For example, there are constructs for report generation and application development. These facilities are beyond the current scope of MLM's. Last, to foster sharing, one wants an open, public standard. There was a danger of using a language that is proprietary and that may be linked to a specific vendor.

Given these considerations it was decided at the conference to design a new syntax for the standard with which to share MLM's. Some of the features from several of the language described above would be considered for inclusion in this standard. Since the conference was being held at the Arden facility, it was decided to name the new standard syntax the Arden Syntax. It was also agreed that the American Society for Testing and Materials (ASTM) should be contacted to serve as the standards body to disseminate this standard when it was complete. A brief overview of the Arden Syntax is described below.

## Arden Syntax description

The Arden Syntax for MLM's has been designed specifically to share medical knowledge. There are constructs for database queries that are intended to facilitate mapping them to an institution's patient database; there are constructs and data structures for time; and there are provisions for tracking changes in MLM's and assigning responsibility for the use of an MLM in an institution. The syntax is derived largely from HELP, which resembles Pascal. In addition, several features in RMRS have been incorporated; a number of its aggregation operators are included, and the manner in which the result of a database query can be manipulated were adopted. Although other systems like Quick Medical Reference (QMR) [7] and DXplain [8] have been assessed for the potential of expressing their knowledge bases in the Arden Syntax, they have not yet been incorporated into the syntax.

Refer to Fig. 1 for a sample MLM. It is derived from a previously published sample MLM [10–13]. The minor differences that have occurred in the syntax reflect insights gained from testing and implementing the syntax over the past year (an example is given in a subsequent section).

### *Basic structure*

An MLM is a text file that is arranged into discrete slots, much like HELP and RMRS. A slot is composed of a slot name (e.g., 'title:') and a slot body. Depending on the slot, the slot body may contain free text, a coded term, or structured data. In order to make the MLM more readable, the slots are grouped into three categories: maintenance, library, and knowledge.

### *Maintenance category*

The maintenance category contains those slots that specify maintenance information that is unrelated to the medical knowledge in the module. The title, filename, author, and institution slots serve to name the MLM and identify its source.

The specialist slot names the person in the institution who is responsible for validating and installing the MLM in the institution. This person sets a validation level in the validation slot: production, research, testing, or expired. The specialist slot should always be blank when transferring MLM's from one institution to another; it is the borrowing institution's responsibility to fill this slot and accept responsibility for the use of the MLM.

Simple version control is accomplished using the version and date slots. Updates to the MLM are tracked using a version number, the date of creation of the MLM, and the date of last revision. As MLM's are revised, old versions are archived.

### *Library category*

The library category contains those slots pertinent to maintenance that are related to the module's knowledge. The purpose slot serves as a location comments. The keywords slot contains descriptive words used for searching through modules and for searching through the literature. Unified Medical Language System (UMLS) terms [17] are preferred are but not mandatory.

References are listed in the citations slot. Comments can appear anywhere in the MLM; those that contain only a number (e.g., '/*4*/') refer to the citation of that number. Links to other sources, such as an electronic textbook, teaching cases, or educational modules may be placed in the links slot.

### *Knowledge category*

The knowledge category contains the actual medical knowledge of the MLM. The type slot identifies the way in which the MLM is used. The data slot defines terms used in the remainder of the MLM. A short name (i.e., a local variable) is substituted for a longer database query.

The evoke slot contains the conditions under which the MLM becomes active. One can view it as setting the context for the MLM. For example, the storage of particular data elements in the patient database may evoke the MLM. The logic slot contains the actual medical rule or medical condition to test for. This slot can contain complex calculations. The action slot specifies what is to be done if the premise is satisfied. Possibilities include sending an alert to a destination, evoking other MLM's and returning data values.

The maintenance and library sections help maintain the MLM. The knowledge section contains the actual logic. The evoke slot specifies that this MLM is evoked whenever an absolute neutrophile count (ANC) is stored. In the data slot, the local variable 'and' is assigned the patient's last 2 ANC'S within the past week. 'pt_taking_tms' is assigned true or false depending on whether the patient is taking trimethoprim and sulfamethoxazole (TMS). The logic slot says that if the patient is taking TMS and if the last ANC is less than 1000 and if the ANC is decreasing, then execute the action slot, which writes the appropriate message. Write 'Caution: The patient's relative granulocytopenia may be exacerbated by trimethoprim/sulfamethoxazole'.

> The syntax of the data, evoke, logic, and action slots is similar to Pascal. For example,
> 3 + 4*2

is a numeric expression (it equals 11). Figure 1 contains other examples in the logic slot.

The Arden Syntax follows RMRS's use of control statements. RMRS uses IF-THEN statements and an EXIT instruction, but it lacks constructs for looping and branching [4]. The result is a more declarative syntax that is easier to implement and that can be easier to read by thos unfamiliar with computer languages. Although HELP does include looping constructs, they are used primarily for result reporting and application development rather than for medical decision logic.

The Arden Syntax includes database queries, constructs for time, and lists, all of which are discussed in the next section.

## Database queries

### *Why queries are essential*

'The use of dioxin in the setting of hypokalemia may cause cardiac arrhythmias' is a simple rule. But applying this rule to a particular patient is difficult. Given the rule and a patient's chart, a physician will

```
maintenance:
  title: Agranulocytosis and Trimethoprim/
  Sulfamethoxazole;
  filenames: anctms;
  version: 2.00;
  institution: Zoo University;
  author: Dr. Bonzo;
  specialist: ;
  date: 7/20/1989, 12/23/1992;
  validation: testing;
library:
  purpose: Display the Arden Syntax;
  keywords:
      granulocytopenia; agranulocytosis;
      trimethoprim; sulfamethoxazole;
  citations:
      1. Anti-infective drug use in relation to the risk
      of agranulocytosis and aplastic anemia. . . Arch
      Int Med 1989; 149 (5): 1036–40.
  links:
      MeSH agranulocytosis/ci and sulfamethoxazole/
      ae;
  knowledge:
  type: data-driven;
  date:
      storage_of ANC: = event {storing of ANC};
      /*neutrophile count in #/mm3*/ and: = read last 2
      from ({query for ANC} where it occurred within
      the past 1 week);
      pt_taking_tms: = read exist {query for TMS
      order};
  evoke: storage_of_ANC;
  logic:
      if pt_taking_tms /*1*/ and last anc < 1000 and
      decrease of anc > 0 then conclude true else
      conclude false endif;
  action: write 'Caution: The patient's
      relative granulocytopenia may be exacerbated
      by trimethoprim/sulfamethoxazole.';
```

*Fig. 1.* A sample MLM.

use common sense when interpreting laboratory values. If the last serum potassium is hemolyzed, the physician will look for the previous one. If the last serum potassium was performed four years ago, then the physician will not use that value. If the value for a recent potassium is 56.6 millimoles/liter, then the physician will suspect that there has been a laboratory error.

Many decision-support systems are able to collect data from a computerized patient database without human intervention (e.g., HELP, RMRS, and ALERTS [15]). This ability obviates the need for health care providers to re-enter data that is already stored electronically; it permits MLM's that check for contraindicators to run in the background without disturbing a health care provider until an alert is actually generated; and it supports the triggering of MLM's without a specific request for assistance [18]. Unfortunately, since such systems lack a human being to filter queries, their MLM's must foresee every contingency for every query to the database.

This section focuses on a critical part of the Arden Syntax: database queries. For each feature in the syntax of the queries, two factors are important: whether the feature is useful to medical rules and whether the feature can be implemented on patient databases in existing clinical information systems. To assess usefulness, several automated decision-support systems were reviewed. Those that use modular independent rules and that are linked to an active patient database were selected for the most detailed study. These included HELP [2, 16], RMRS [4, 19], and ALERTS [15]. To assess the feasibility of the query syntax, several well-known clinical information systems were reviewed: HELP, RMRS, Computer Stored Ambulatory Record (COSTAR) 820], The Medical Record (TMR) [21], Summary Time Oriented Record (STOR) [22], Patient Care System (PCS) [23], and the Beth Israel Hospital and Brigham and Women's Hospital clinical computing systems [24, 25].

### *Disparate vocabularies and database structures*

One of the more difficult problems of sharing an MLM between two institutions is the difference in the vocabularies and the database structures used in the institutions [10–13]. Most institutions' vocabularies are a unique mixture of local nomenclature, terms used by vendors in purchased systems, and standard vocabularies like ICD-9. The review of the clinical information systems revealed such diversity in vocabulary and database structure [2, 19–25] that the definition of a single syntax for querying patient data bases was not attempted. The UMLS [17] has made progress in mapping between standard vocabularies, but automatically translating queries to accommodate disparate patient databases and mapping terms to local clinically descriptive vocabularies is probably not feasible at the current time.

Instead, queries are split into two components: an institution-specific component phrased in the authoring institution's own query language, and a more general component phrased in the Arden Syntax. Then MLM's are shared, the receiving institution must manually translate the institution-specific component into its own query language.

Following the example of HELP [16] and RMRS [4], access to institution-specific data has been extracted from the logic and placed in a separate data slot. This serves to isolate the part of an MLM that will generally need to be changed when an MLM is shared. Hopefully these MLMs will need little alteration. The degree to which logic can be written in multiple centers and shared among several institutions has been the subject of several studies (e.g., [26, 27]) and the results so far are promising. Within the data slot, institution-specific queries are mapped to local names used elsewhere in the MLM. As long as the local names have been assigned the same data elements, the logic of the MLM may undergo little change despite major changes to the queries.

*Availability of data*

If an MLM requires a data element that is not available in an institution's database, then the MLM cannot be used without modification. Sometimes a missing data element can be derived from other data that are stored in the database. For example, an MLM that requires a patient's systemic vascular resistance can be used by an institution that records mean arterial pressure and cardiac output, from which the systemic vascular resistance can be calculated. (SVR = MAP/CO.)

The data slot has provisions to perform calculations on the results of queries. Then the result of the calculation can be assigned to the local variable name used in the logic of the MLM. In this way, the logic need not know whether the data came from a direct query or a calculation on a set of queries.

*Aggregation operators*

An aggregation operator performs a summary function on a set of data based on some property of the data; examples include MAXIMUM, LAST, and SUM. Aggregation has been found to be useful in medical rules in HELP [16], RMRS [4], and ALERTS [15].

A database management system (DBMS) can support aggregation in several ways. First, it can provide aggregation directly through its own query language, as do relational databases that use SQL [28]. Second, it can provide aggregation indirectly by supplying primitive operations that can be assembled to perform aggregation in an efficient manner. A DBMS that can retrieve the last occurrence of a patient attribute and then iteratively retrieve preceding instances is sufficient to perform all the aggregation operations in the Arden Syntax except FIRST. Third, the aggregation operator can be applied by support routines after the query is completed. This third alternative is not feasible for operators like LAST for efficiency reasons; since one would not want to retrieve all instances of some patient attribute just to use the last one. All of the clinical information systems reviewed appeared to support at least some form of aggregation [2, 19–25]. It is likely that missing operators could be added to these systems by writing the appropriate support routines.

There are relatively few aggregation operators, so agreeing on a common syntax for them is not as difficult as it is for general medical terms. The aggregation operators have been separated out from the rest of the query, and a standard syntax has been defined for them.

*Time*

HELP [16], RMRS [4], and ALERTS [15] use a temporal model that supports discrete points in time (e.g., 'now' or '13 : 12 : 13 1993-01-28), and duration, which are periods of time (e.g., '2 months' or '3 days'). These elements can be put together for form more complex expressions:

3 months ago

4 weeks before now
2 days after the time of surgery

Operators use these elements to create search windows:

. . .where it occurred within the past 3 months

All of the patient databases that were reviewed stored at least discrete time points along with the data elements, and should be capable of supporting the operations in the model [2, 19–25].

More sophisticated models of time were considered. (For example, STOR records discrete time values along with an associated value that indicates the precision with which the time value is known [22].) But because of its common usage, simplicity of design, ease of implementation, and success in the above mentioned decision-support systems, the discrete temporal model is used in the Arden Syntax.

Like aggregation operators, the time constraints have a limited vocabulary, and mapping them to an institution's query language need not be difficult. They, too, have been separated from the remainder of the query, and a standard syntax has been defined for them.

*Query syntax*

The general form of a query is:

<var>: = READ<aggregation>({<body>}
WHERE<t_constraint>)

where <var> is a local variable; <aggregation> is one or more aggregation operators; <body> is the institution-specific database query devoid of aggregation and time constraints; and <t_constraint> is the temporal search window.

For example:

k: = READ THE MAXIMUM ({select potas from electro} WHERE IT OCCURRED WITHIN THE PAST 3 DAYS);

The local variable 'k' is assigned the maximum serum potassium within the past three days. 'MAXIMUM' is an aggregation operator that picks the highest potassium value. The phrase 'select potas from electro' is the institution-specific body of the query, devoid of aggregation and time constraints, written in the query language of the institution's DBMS (a relational database in this example). 'WHERE IT OCCURRED WITHIN THE PAST 3 DAYS' is the temporal search window for the query.

When an institution wants to run an MLM, it must compile the high-level form into a form understandable by its computer system. A part of the compilation process is to insert the aggregation and time constraints into the body of the query using the institution's query language. For example, a compiler would automatically turn the last query into this:

k: = READ {select max(potas) from electro where time> (now – 3 days)};

assuming that 'time' is a valid column and 'days' is a valid term in the query language. This new body could then be executed by the institution's DBMS. This version of the query need not be altered or even seen by humans.

When another institution acquires the MLM, someone would have to alter it manually to fit their own patient database. If the institution uses a completely foreign database structure, the altered query may look like this:

k: = READ THE MAXIMUM ({123 from 456 to 460} WHERE IT OCCURRED WITHIN THE PAST 3 DAYS);

Although the institution-specific body changes, the aggregation operators and the time constraint need not be changed. The institution's compiler would then insert the aggregation and time constraints into the body at the time of compilation.

To determine whether it is possible to separate out the aggregation and time constraints and then to automatically insert them depends on the DBMS. So far two systems have been studied in detail. A relational database which uses SQl and a hierarchical database which uses IMB's DL/I query language along with a set of support routines. A compiler that automatically inserts these elements is operational for both of these systems.

A question remains as to whether other systems will be able to do the same. At worst, the entire query can be written in the institution's query language when aggregation and time constraints cannot be separated logically from the rest of the query. This capability creates more work on transferring the MLM, but it accommodates a wider range of DBMS's.

*Lists*

The results of these queries are used in the logic slot. Unless specific aggregation operators are used, one cannot know how many items will be returned by a query. For example, even if a query requests a patient's age, multiple values may be returned because of conflicting evidence stored in the database.

Both RMRS and HELP allow several items to be returned as the result of a query: RMRS does so by default, and HELP uses its 'relation' construct. Similarly, the Arden Syntax allows local variables to hold lists of items. For example:

potas: = {select potas from electro}

would place all the serum potassiums a patient had ever had in the local variable 'potas'.

The resultant list can be manipulated in the logic slot using aggregation operators much like the ones in database queries. Continuing with the above example, the following expression in the logic slot:

MAXIMUM potas/AVERAGE potas

would result in the ratio between the largest potassium value in the list and the average potassium value. The selection operator, WHERE, can also be applied to lists. The following expression:

potas WHERE T> 5

would select only those potassium values that are greater than 5.

**Status of Arden Syntax**

A new subcommittee of ASTM (31.15) was formed with T. Allan Pryor as chairman and George Hripcsak as secretary. This subcommittee was charged with the responsibility of creating standards for Medical Knowledge Representation. In April of 1992 the work of the subcommittee culminated in the publication of ASTM Standard E 1460 a Standard Specification for Defining and Sharing Modular Health Knowledge Bases (Arden Syntax for Medical Logic Systems). This standard is now available for all interested parties and can be obtained by contacting ASTM [34].

During the development of the standard 17 University/Vendor groups have shown interest in the work and are actively developing compilers to implement the Arden Syntax on their systems. Among these 17 groups 9 are academic sites located both in the United States and Europe. Eight vendors of hospital information systems have begun either actively creating MLMs or are evaluating the use of MLMs as an adjunct to their existing systems.

Columbia University has agreed to serve as the repository for MLMs. Sites developing MLMs are asked to submit their knowledge bases to Columbia. Columbia will then catalog and store the MLMs. Sites wishing to receive copies of existing MLMs can then contact Columbia. To date Columbia has received MLMs from 5 institutions. Anyone wishing to obtain those examples should contact George Hripcsak at the Center for Medical Informatics, Columbia Presbyterian Medical Center, New York City, New York 10032.

**Conclusion**

The routine sharing of medical knowledge bases is a large and very important project. By limiting the scope of the project to modular independent knowledge and by basing it on existing systems, such sharing of knowledge may become feasible. The Arden Syntax is designed to facilitate sharing medical knowledge. The knowledge is contained in independent units called MLM's. They include information necessary to track and maintain the knowledge base. The syntax for database queries is intended to ease the transfer of MLM's between institutions. Features that have been found to be useful in existing, decision-support systems have been incorporated into the queries, and it appears that clinical information systems will be able to accommodate them. Perhaps most important, the Arden Syntax is a public standard, and a new ASTM subcommittee is responsible for the syntax.

Working systems that use the standard have been and are being constructed. Copies of the complete syntax can be obtained from ASTM [34].

## References


1. Bleich HL. Computer Evaluation of Acid-Base Disorders. J Clinical Invest 1969; 48: 1689–93.
2. Pryor TA, Gardner RM, Clayton PD, Warner HR. The HELP system. In: Blum BI, editor. Information Systems For Patient Care. New York: Springer Verlag, 1984: 109–28.
3. Kuperman GJ, Gardner RM, Pryor TA. HELP: A Dynamic Hospital Information System. Springer-Verlag Inc: New York NY, 1991.
4. McDonald CJ. Action-Oriented Decisions in Ambulatory Medicine. Chicago: Year Book Medical Publishers, 1981.
5. McDonald CJ. Protocol-Based Computer Reminders, the Quality of Care and the Non-Perfectibility of Man. N Eng J Med 1976; 295: 1351–5.
6. McDonald CJ, Tierney WM, Overhage JM, Martin DK, Wilson GA. The Regenstrief Medical Record System: 20 Years of Experience in Hospitals, Clinics, and Neighborhood Health Centers. MD Comp 1992; 9(4): 206–17.
7. Miller RA, McNeil MA, Challinor SM, Masarie FE, Myers JD. The Internist-1/Quick Medical Reference Report. West J Med 1986; 145: 816–22.
8. Shortliffe EH. Computer-Based Medical Consolation: MYCIN. Elsevier: North Holland, New York, 1976.
9. Barnett GO, Cimino JJ, Hupp JA, Hoffer EP. DXplain: An evolving diagnostic decision-support system. JAMA 1987; 258(1): 67–74.
10. 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, editor. Proc 13th SCAMC, IEEE Comp Soc Press, 1989: 116–21.
11. Clayton PD, Pryor TA, Wigertz OB, Johnson SB, Hripcsak GM. Sharing medical knowledge for automated decision-making. In: Greenes RA, editor. Proc 12th SCAMC, IEEE Comp Soc Press, 1988: 591–4.
12. Wigertz OB, Clayton PD, Hripcsak G, Linnarsson R. Knowledge representation and data model to support medical knowledge base transportability. In: Talmon J, Spiegelhalter J, editors. System Engineering in Medicine. Heidelberg: Springer Verlag, 1989.
13. Yasnoff WA. Technology transfers of medical decision assistance systems. In: Hammond WE, editor. AAMSI Congress 89, AAMSI Publishers, 1989: 103–7.
14. Warner HR. The Iliad Program: An expert computer diagnostic program. J Med Prac Manag 1992; 8(2): 123–8.
15. Shabot MM, LoBue M, Leyerle BJ, Dubin SB. Inferencing strategies for automated ALERTS on critically abnormal laboratory and blood gas data. In: Kingsland LC, editor. Proc 13th SCAMC, IEEE Comp Soc Press, 1989: 54–7.
16. HELP Frame Manual, Version 1.6. Salt Lake City: LDS Hospital, 1989.
17. Lindberg DAB, Humphreys BL. Toward a unified medical language. Medical Informatics Europe 87, Proceedings of the Seventh International Congress, Rome, 1987; 1: 23–31.
18. Clayton PD, Evans RS, Pryor TA, Gardner RM, Haug PJ, Warner HR. Data driven interpretation of laboratory results in the context of a medical decision support system. In: Keller H, Trendelenburg C, editors. Data Presentation/Interpretation. Berlin: Walter de Gruyter, 1989: 367–80.
19. McDonald CJ, Blevins L, Tierney WM, Martin DK. The Regenstrief Medical Records. MD Computing 1988; 5(5): 34–47.
20. Barnett GO, Zielstorff RD, Piggins J, McLatchey J, Morgan MM, Barrett SM et al. A comprehensive medical information system for ambulatory care. In: Blum BI, editor. Proc 6th SCAMC, IEEE Comp Soc Press, 1982: 8–18.
21. Stead WW, Hammond WE. Computer-based medical records: The centerpiece of TMR. MD Computing 1988; 5(5): 48–62.
22. Whiting-O'Keefe QE, Whiting A, Henke J. The STOR clinical information system. MD Computing 1988; 5(5): 8–21.
23. Patient Care System ORDERS. Database Reference Manual. Atlanta: International Business Machines Corporation, 1988.
24. Safran C, Slack WV, Bleich HL. Role of computing in patient care in two hospitals. MD Computing 1989; 6(3): 141–8.
25. Safran C, Porter D, Rury CD, Herrmann FR, Lightfoot J, Underhill LH et al. ClinQuery: Searching a large clinical database. MD Computing 1990; 7(3): 144–53.
26. Giuse NB, Giuse DA, Miller RA. Computer assisted multicenter creation of medical knowledge bases. In: Greenes RA, editor. Proc 12th SCAMC, IEEE Comp Soc Press, 1988: 583–90.
27. Lincoln NJ, Turner C, Hesse B, Miller R. A comparison of clustered knowledge structures in Iliad and in Quick Medical Reference. In: Greenes RA, editor. Proc 12th SCAMC, IEEE Comp Soc Press, 1988: 131–5.
28. Date CJ. An Introduction to Database Systems, volume 1. Reading: Addison-Wesley Publishing Company, 1986.
29. Lesk ME, Schmidt E. Lex – a lexical analyzer generator. Computer Science Technical Report No. 39. Murray Hill: Bell Labs, 1975.
30. Johnson SC. Yacc: Yet another compiler. Computer Science Technical Report No. 32. Murray Hill: Bell Labs, 1975.

31. Tremblay J-P, Sorenson PG. The Theory and Practice of Compiler Writing. New York: McGraw-Hill Book Co, 1985.
32. Hripcsak G, Clayton PD, Cimino JJ, Johnson SB, Freidman C. Medical Decision Support at Columbia-Presbyterian Medical Center. IMAI Working Conference on Software Engineering in Medical Informatics, Amsterdam, 8–10 October, 1990.
33. 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. IMIA Working Conference on Software Engineering in Medical Informatics, Amsterdam, 8–10 October, 1990.
34. ASTM, 1916 Race Street, Philadelphia, PA 19103-1187.

*Address for offprints:*
T.A. Pryor,
Med. Biophysics & Comput,
University of Utah,
LDS Hosp./325 Eighth Ave.,
Salt Lake City, UT 84143, U.S.A.