MEDINFO 2004
M. Fieschi et al. (Eds)
Amsterdam: IOS Press

Medical Expert Systems Developed in j.MD, a Java Based Expert System Shell Application in Clinical Laboratories.

Viviane Van Hoof [a], Arno Wormek[b], Sylvia Schleutermann[b], Theo Schumacher[c], Olivier Lothaire[c] Christian Trendelenburg[d].

[a]Dept. of Clinical Chemistry, University Hospital Antwerp, Belgium,
[b]Wormek – Information Science and Technology in Medicine, Uting, Germany, [c]Sysmex Belgium SA, Barchon, Belgium
[d]Institute for Laboratory Medicine – University of Frankfurt - Städtische Kliniken Frankfurt a.M. – Höchst, Germany

Abstract

Growing complexity of diagnostic tests, combined with increased workload, stringent laboratory accreditation demands, continuous shortening of turn-around-time and budget restrictions have forced laboratories to automate most of their iterative tasks. Introduction of artificial intelligence by means of expert systems has gained an important place in this automation process. Different parts of clinical laboratory activity can benefit from their implementation and the present project deals with one aspect, namely the clinical interpretation of diagnostic tests. This paper describes how j.MD, a new Java based expert system shell, was used to reprogram the expert system for interpretation of amylase isoenzyme patterns that has been in use for many years in our laboratory, and that was originally programmed in Pro.MD, a Prolog based expert system shell. One of the most important advantages of the j.MD system is its bidirectional link with the laboratory information system. This project shows how expert systems for the interpretation of complex diagnostic tests that demand specific expertise can become an integrated part of the automated clinical chemistry lab.

Keywords :

knowledge management, decision support systems, expert systems, Java, isoenzyme patterns

Introduction

Due to the continuing automation of clinical laboratory tests, many examinations that used to be technically complicated and time consuming become less elaborate, and more routine clinical laboratories are/will be able to perform these tests. However, the amount of test data that are generated and their complexity, make it impossible for laboratories to acquire all the necessary 'in house' expertise for their clinical interpretation. When present, the expert is not always available, and even a trained expert can find it difficult to interprete reports always in the same standardized way. On the other hand there is an increasing awareness that the responsibility of specialists in laboratory medicine goes much deeper than merely producing analytical data : they should be advisors in test selection, test logic and test interpretation (Mission Statement of the American Association of Clinical Chemists, 1995). Expert systems for the interpretation of diagnostic tests are potential tools to solve this dilemma.

Two expert systems for the interpretation of alkaline phosphatase and amylase isoenzyme patterns, programmed in Pro.MD, a rule based expert system shell written in Prolog, have been developed in our laboratory [1-3]. After extensive testing and a thorough medical validation process, they were introduced for routine use in 1988 and 1997 respectively. The systems were regularly updated since (11 and 5 times respectively). Updates were triggered by user feedback and/or the availability of new scientific data. By now, more than 19 000 alkaline phosphatase and more than 2 000 amylase isoenzyme patterns have been evaluated and the number of requests by in-house and external clinicians has been increasing steadily. This proves that it is feasable to successfully introduce expert systems in a routine clinical laboratory. However, a serious draw-back of Pro.MD is the lack of a bidirectional link with the laboratory information system (LIS). As a consequence, lab results have to be introduced manually in the system or via a unidirectional download and the results of the expertise cannot be integrated in the electronic laboratory report of the patient. These inconveniences stood in the way of a more general use of these systems. To solve this problem we decided to look for a new more user-friendly tool wherein we could re-introduce the knowledge that was integrated in the knowledge base (KB) of the previous systems. To accomplish this goal, the expert system shell had to answer the following demands : 1° bidirectional communication with the LIS, 2° universal platform, 3° flexibility and user-friendliness.

Materials and methods

j.MD (*J*ava-System supporting *M*edical *D*iagnostic Reasoning), a Java-based environment that answered the requirements mentioned above, was selected as our new tool [2,4]. As a first step, we reprogrammed the knowledge base for the interpretation of amylase isoenzyme patterns in j.MD. A bidirectional link between the LIS and the expert system was realized by our LIS provider Sysmex Belgium (Barchon, Belgium).

Technical specifications :

Bidirectional communication between Molis and j.MD

Communication between Molis and j.MD is set up in a TCP/IP client-server socket architecture. Data and results are transmitted as XML documents. On the j.MD side, a server is started that listens to a specific port number. On the Molis side, a connection is established to the j.MD Server through a client socket connection on the specified IP-address for the server and the specified port number. Test results are identified by their LOINC© (*L*ogical *O*bservation *I*dentifiers *N*ames and *C*odes) codes, published by the Regenstrief Institute and public domain. LOINC© is currently being established as a standard for uniquely identifying laboratory tests [5]. The Molis datastore contains a mapping for associating laboratory tests with their proper LOINC© code. When the j.MD Server receives data from the Molis client, it extracts the order data as an XML document and passes it on to the j.MD Runtime module for processing. The j.MD Runtime adds all its results to the original document. This document, augmented with the j.MD results, is sent back to the Molis client by the j.MD Server. The Molis client then extracts the relevant results and reports and integrates the data back into the Molis datastore. Further handling of the results is done by Molis, for example integrating report text into Molis reports. The expertise that is added to the order document by the j.MD Runtime module contains interpretations for test results into categories such as "reduced" or "normal", suggestions for further tests, diagnostic hints and suspected diagnoses.

Technical Details on j.MD

The j.MD software is written in Java™ to assure platform-independence and portability. The version of the Java Virtual Machine that the software is released with is the JVM 1.3.1 from Sun Microsystems. The j.MD datastore is loaded and saved as an XML document to assure flexibility and multi-access use. Exchange of j.MD Knowledge Bases is also based on XML documents so that knowledge base authors can easily distribute their work. All data communication is based on XML documents. This facilitates the use of j.MD in client-server environments as well as web environments. The j.MD software consists of several modules plus a set of public domain software packages which are described below. In principal, the software runs on any machine for which a Java Virtual Machine (preferrably version 1.3.1) exists. However, in non-Win32 environments, the LOINC© database, the license code checker and the installer would have to be imported.

j.MD modules :

j.MD Datastore and Utility Base

The j.MD Datastore and Utility Base consist of the basic data structures to represent a repository of knowledge bases assembling knowledge base objects, such as parameters, formulas, interpretations, concepts, rules, patient models and text patterns. In addition, corresponding to the objects in the repository, there are so-called builder classes that construct the object from configuration options. The utility base consists of service classes and functions that perform transformations (for example unit transformations), that load and save repositories and that import and export knowledge bases into / from remote repositories. The remote repositories can reside on any computer that is accessible from the j.MD Datastore and Utility Base by URL (Uniform Resource Locator). This module is the base for all further j.MD modules.

j.MD Knowledge Editor

The j.MD Knowledge Editor is the workbench for the knowledge base author. Knowledge bases and their objects are created through wizards and can be edited and refined in designated editors.

j.MD Report Generator

The j.MD Report Generator creates reports from text patterns defined by the knowledge base author. The reports can be styled and the currently supported formats for styled reports is HTML (for styled web based reports) and XML-FO (Formatting Objects) for reports that can be printed or saved for example as PDF reports or RTF reports.

j.MD Runtime

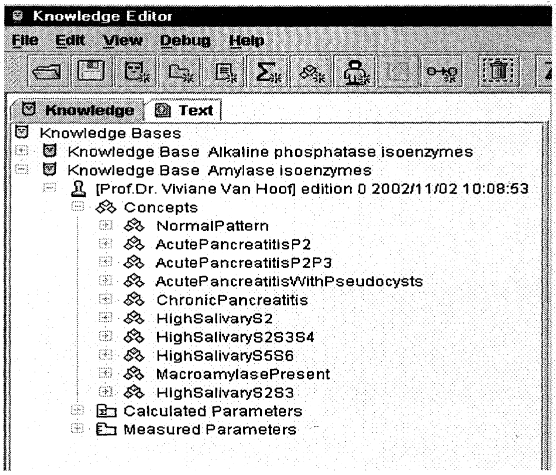
The j.MD Runtime is a module that accepts XML documents with order data (patient and requester identification, test results and entry diagnoses), processes it and returns it to the caller. All results are re-integrated into the order document. The j.MD Runtime is multi-threaded, which means that it can operate in a web environment with many simultaneous client requests. To process the results, the j.MD Runtime first sets up the runtime session with all relevant knowledge base objects. Knowledge base objects such as rules are compiled only by request to optimize compile time. The runtime session stays static over all client requests. Each client request then operates in its own runtime scope with the common (static) runtime session being the top-level scope. This architecture assures that all clients work within their environment of test results and j.MD results, but still, compiled knowledge base objects are available to all clients.

j.MD Debugger

The j.MD Debugger is a tool for the knowledge base author to test all aspects of a knowledge base. The debugger consists of several parts, reflecting the debug workflow. In the first part, the user selects all items of the knowledge base that he or she wishes to test. Selectable parts are: knowledge bases, concepts, rules and text patterns. This enables the user to analyze, for example, the behavior and performance of a specific rule. In the second part, the user enters input data to test the selected knowledge base objects. The user can trigger the creation of editors for data entry. Entered data can be saved and reloaded later as test cases. This aids the author when refining an already existing knowledge base. In the third part, the user can start the j.MD Runtime with an order document created from the debug targets and debug inputs. The j.MD Debugger will receive the resulting document from the j.MD Runtime and will display all results in the corresponding panes. In addition, the user can review the rule trace. In the fourth part, the user can preview and print the selected reports and report fragments.

j.MD Front End

When there is no link with the LIS, the user can manually enter patient demographics, test results and entry diagnoses using the j.MD Front End module. The front end assembles the data into a proper order document and starts the j.MD Runtime to process it. The results are displayed either in a so-called j.MD Standard Report, or in a report specifically configured for the relevant knowledge base.

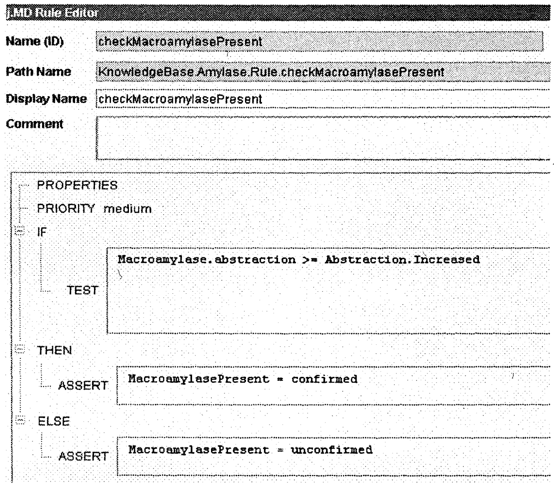


Knowledge Bases
- Knowledge Base Alkaline phosphatase isoenzymes
- Knowledge Base Amylase isoenzymes
  - [Prof.Dr. Viviane Van Hoof] edition 0 2002/11/02 10:08:53
    - Concepts
      - NormalPattern
      - AcutePancreatitisP2
      - AcutePancreatitisP2P3
      - AcutePancreatitisWithPseudocysts
      - ChronicPancreatitis
      - HighSalivaryS2
      - HighSalivaryS2S3S4
      - HighSalivaryS5S6
      - MacroamylasePresent
      - HighSalivaryS2S3
    - Calculated Parameters
    - Measured Parameters

Figure 1 - Concepts that were defined in the j.MD isoamylase knowledge base

j.MD Server

The j.MD Server is a TCP/IP-based socket server for bidirectional communication with data providers such as for example, a LIS, a database client or a web client. The general architecture and functionality were described in Part 1.



Name (ID) checkMacroamylasePresent

Path Name KnowledgeBase.Amylase.Rule.checkMacroamylasePresent

Display Name checkMacroamylasePresent

Comment

```
PROPERTIES
PRIORITY medium
IF
    TEST    Macroamylase.abstraction >= Abstraction.Increased
THEN
    ASSERT  MacroamylasePresent = confirmed
ELSE
    ASSERT  MacroamylasePresent = unconfirmed
```

Figure 2 - Example of a rule in the KB module checking the presence of a macroamylase fraction.

Apache modules

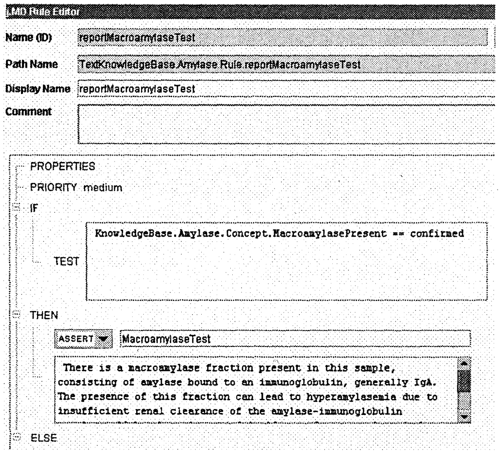
Apache modules are mainly used for processing XML documents (parsing, transforming, processing XML search queries). In addition, the j.MD software uses the Apache FOP module for processing XML reports that are styled with XML-FO (Formatting Objects).

Rhino project

The JavaScript engine of the Rhino project of Mozilla is used to implement the rule engine in the j.MD Runtime.

LOINC©

As described above, the j.MD software uniquely identifies laboratory tests by their LOINC© codes. The Regenstrief Institute deploys an MS ACCESS database of LOINC© codes which is integrated into the j.MD software via a LOINC© database viewer, offering support for finding the proper LOINC© code for a specific laboratory test. The database viewer is not part of the Regenstrief Institute's distribution.



Name (ID) reportMacroamylaseTest

Path Name TextKnowledgeBase.Amylase.Rule.reportMacroamylaseTest

Display Name reportMacroamylaseTest

Comment

```
PROPERTIES
PRIORITY medium
IF
    TEST    KnowledgeBase.Amylase.Concept.MacroamylasePresent == confirmed
THEN
    ASSERT  MacroamylaseTest
     There is a macroamylase fraction present in this sample,
    consisting of amylase bound to an immunoglobulin, generally IgA.
    The presence of this fraction can lead to hyperamylasemia due to
    insufficient renal clearance of the amylase-immunoglobulin
ELSE
```

Figure 3 - rule in the text module defining the text pattern that will be integrated in the patient report when the presence of a macroamylase fraction was confirmed by the rule shown in Figure 2.

Results

A prototype version of the j..MD system for the interpretation of amylase isoenzyme patterns has been developed. Like the older version, the system takes into account demographic data such as age and gender of the patient, and results of other laboratory tests that are relevant for the performed examination, such as liver and renal function tests. Two new parameters were introduced : serum lipase and γ-glutamyltransferase activities. Cut-off values for the different fractions in health and disease are based on previous studies [6-8]. Lower and upper limits of normal were determined in a healthy reference population of 396 adults (aged 20 –65 years) and 986 children (aged 4-19 years). Studies were performed in patients with acute and chronic pancreatitis, pre- and post ERCP, malignant pancreatic tumours and parotitis. Prior to its implementation in 1997, the system was extensively validated in cooperation with the gastro-enterologists of the hospital. Concepts that were implemented in the prototype j.MD version are shown in Figure 1. An example of a rule and the rule defining the corresponding text pattern are shown in Figures 2 and 3 respectively. The bidirectional link has been tested within a mirror LIS database on a separate server and is ready to be implemented in real time. Testing of the expert system with 'dummy' data and real patient data has been performed and the system has been updated accordingly. The system's performance is now being critically evaluated by the clinicians (end-users). After a final update the j.MD expert system for the interpretation of amylase isoenzyme patterns will be implemented in the routine clinical laboratory. Figure 4 shows how in our laboratory the Pro.MD / j.MD systems for the interpretation of diagnostic tests complement QC-Today, developed in collaboration with IL (Zaventem, Belgium) [9], and VALAB (EREMS Medical, Toulouse, France), two other bidirectionally linked expert systems for QC and medical validation that have been in routine use since 2001 and 1998 respectively.

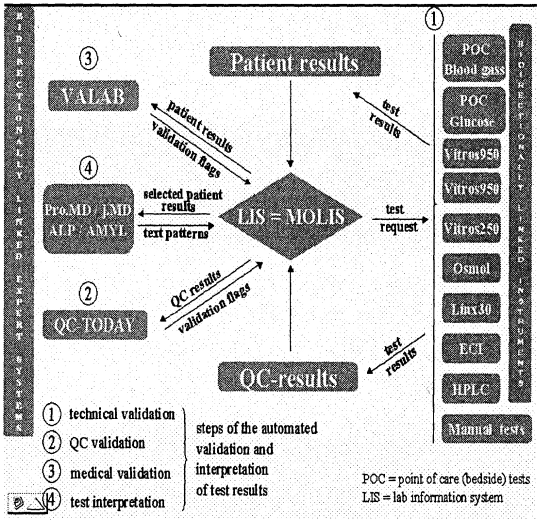


Figure 4 - flow-chart showing how expert systems have become an integrated part of the automated routine clinical chemistry lab. VALAB : medical validation program developed by Valdigué (Toulouse, France) ; QC-Today : QC validation program developed in collaboration with IL (Zaventem, Belgium)

Discussion

The need to embed expert systems in the LIS is not new [10], and an object-based approach seems an appropriate way to achieve this goal [2,11]. Nevertheless, most expert systems for clinical test interpretation still depend on manual data input or a unidirectional download from the LIS. This paper describes how a Java based platform can be used to realize a bidirectional link, which is less error-prone and reduces workload considerably.

A correct clinical interpretation of amylase isoenzyme patterns is time consuming and quite complicated, therefore routine laboratories tend to send samples for these tests to reference laboratories, or they simply do not perform them any more ("orphan test"). Nevertheless this relatively simple and cheap analysis provides a treasure of relevant clinical informations, e.g. regarding the pancreatic function of the patient, and can avoid the performance of more invasive and costly examinations. Isoamylase patterns are of particular interest post-ERCP, where determination of the isoenzyme pattern helps to differentiate between a rise of total amylase activity due to post-procedure pancreatitis or due to an irritation of the parotid glands following passage of the endoscope. Amylase is a small molecule that is readily filtered by the normal kidney, but when the enzyme binds to an immunoglobulin, the resulting macromolecule cannot be effectively cleared and serum amylase activity rises. It is important to distinguish benign macroamylasemia from other clinically important causes of an elevated total amylase activity that require further, sometimes invasive, examinations of the patient. These are only two examples of the clinical importance of a correct interpretation of amylase isoenzyme patterns.

The same applies to alkaline phosphatase isoenzyme patterns, a far more complex domain that will be the next Pro.MD knowledge base to be re-programmed in j.MD.

Conclusion

More than 15 years of experience with Pro.MD have shown that the routine use of expert systems for the interpretation of amylase and alkaline phosphatase isoenzyme patterns adds considerable value to the mere dosage of total enzyme activity. The present project shows that using j.MD as expert system shell and linking it bidirectionally with the LIS offers the possibility to overcome the drawbacks of the older systems and to obtain in this way a complete integration of clinical interpretation of complex diagnostic tests within the lab automation system (Figure 4).

Acknowledgments

We are indebted to Roland Schnepel from Sysmex Europe (Norderstedt, Germany) for making the j.MD expert system shell available for the development of the isoamylase and alkaline phosphatase expert systems.

References

[1] Pohl B, Trendelenburg C. Pro.MD – A diagnostic expert system shell for clinical chemistry test result interpretation. *Meth Inform Med* 1988;27 :111-117.

[2] Trendelenburg Chr, Colhoun O, Wormek A, Massey KL. Knowledge-based test result interpretation in laboratory medicine. *Clin Chim Acta* 1998;229-242.

[3] Van Hoof VO, Lepoutre LG, De Broe ME. Development and clinical use of a computer assisted decision support system for the interpretation of alkaline phosphatase isoenzyme patterns. *Lab Med* 1994;18:558-563.

[4] Schumacher T. The Java-Medical Diagnostics expert system shell. *Infus Ther Transfus Med* 2001;28:292-296.

[5] McDonald JC, Huff SM, Suico JG, Hill G, Leavelle D, Aller R, Forrey A, Mercer K, De Moor G, Hook J, Williams W, Case J, Maloney P. LOINC, a universal standard for identifying laboratory observations : a 5-year update. *Clin Chem* 2003;49:624-633.

[6] Van Hoof VO, Geeraerts L, Van Meirvenne M, Van Mullem M, Chevigné R, Van Campenhout C, Lepoutre L. Reference values for amylase isoenzymes determined by agarose electrophoresis. *Clin Chem* 1990;36:1251-1252.

[7] Van Hoof VO, Michielsen P, Roelandt R, Geeraerts L, Van Mullem M, Van Campenhout C, Van Maercke Y, Lepoutre L. Evaluation of the Iso-Amyl system and its application in a prospective study of serum amylase isoenzymes after endoscopy. *Biologie prospective*, Ed M-M Galteau, G Siest, J Henney, 1989, p 261-264.

[8] Michielsen P, Van Hoof VO, Lepoutre L, Van Maercke Y. Serum pancreatic isoamylase activity in pancreatic disease. *Acta Gastroenterol Belgica* 1991;54:164-168.

[9] Van Hoof V, Cluckers H, Verhaege V, Truyers M, Rombouts P, De Groote S, Hek J, Molenaer J, Jones E, Schumacher T. Automation of internal quality control procedures by implementation of a bidirectional communication between QC-Today (IL) and the LIS (Molis). *Clin Chem Lab Med* 2002;40:M-SA-040 (S156).

[10]Conelly DP. Embedding expert systems in laboratory information systems. *Am J Clin Pathol* 1990;94:S7-14

[11]Barsalou T. An object-based architecture for biomedical expert database systems. *Comput Method Programs Biomed* 1989;30:157-168.