Using Ontologies to Build a Knowledge Base of Cardiology Problems and Algorithms

Ch Bratsas[1], P Quaresma[2], G Pangalos[1], N Maglaveras[1]

[1]Aristotle University, Thessaloniki, Greece
[2]University of Evora, Evora, Portugal

Abstract

Whereas cardiology is knowledge-intensive and already well computerized, it is rather difficult to find appropriate algorithms, which solve Medical Informatics Cardiology Problems. In this paper, a knowledge base of problems and algorithms in cardiology via ontologies, as well as the architecture for a manageable and efficient search of appropriate algorithms will be presented and analyzed. The ontologies' construction could standardize the process of describing cardiology problems and algorithms from expert users – doctors and computer scientists respectively. These ontologies along with a set of instances-descriptions will generate informatics cardiology problems and algorithms knowledge base. A specific ontology was built; the MedProAlg ontology. The Cardiology experts and the Informatics experts described 32 Medical Informatics Cardiology Problems and algorithms that solve them correspondingly via MedProAlg.

1. Introduction

Several authors have noted the value of medical algorithms in healthcare. McGinn *et. al.* note that validated clinical decision rules have "the potential to inform clinical judgment, to change clinical behavior, and to reduce unnecessary costs, while maintaining quality of care and patient satisfaction" [1]. Other researchers have noted that "quantitative methods to enhance clinical judgment would be of tremendous benefit to physicians caring for the critically ill" [2].

A plethora of medical information exists in the form of published algorithms. Most clinicians use only a small subset routinely. The obstacles to their use include the lack of knowledge that they exist, uncertainty about their boundaries, difficulty in converting to the units expressed in the algorithm, and lack of availability at the point of care. Algorithms would be more widely used if they were readily available in a practical format to clinicians, educators and researchers [3].

To find and utilize a specific algorithm that will solve a particular problem, one must first know in detail what the problem is. In order to achieve a manageable and efficient search of appropriate algorithms, which will be able to solve medical informatics problems, the creation of a framework, where medical practitioners will describe the problems' structure and informatics practitioners will generate the algorithms that solve these problems, will be suggested. To provide a structure framework for modeling the problems and algorithms in medical informatics two domains of knowledge are required: the domain of medicine and the domain of computer science. Building ontologies, which will offer the possibility to describe medical informatics problems and the algorithms that solve these problems, would possibly provide with an answer to the need of combining, exchanging and reusing knowledge representation of both domains.

Ontology can be defined as an explicit and specific description of domain knowledge [4]. As far as medical ontologies are concerned, the schemas that have already been categorised are mostly repositories of concepts and controlled dictionaries, which provide definitions and semantic relationships among the concepts.

A medical problem shall be called Medical Informatics - Cardiology- Problem (MIP or MICP), if its solution is an algorithm that can be implemented via informatics.

In general, an algorithm is defined as "a step-by-step procedure for solving a problem or accomplishing some end especially by a computer" [5].

In order to accomplish an efficient search of appropriate algorithms in medical informatics a framework able to describe problems and algorithms in medical informatics via ontologies is suggested.

2. MedProAlg ontology

This project implements a prototype ontology, which has been built in Protégé-2000 [6]. This ontology was named as MedProAlg Ontology (Medical Problem Algorithm Ontology). There are two important matters that should be noted before presenting MedProAlg ontology; the first one is the utilization of UMLS during problem categorization (class UMLS_ROOT), and the second one is the fact that the class pragmatics, which

0276−6547/04 $20.00 © 2004 IEEE

describe certain elements of problems and algorithms, has been based on Dublin core elements metadata.

The Dublin Core (DC), conceived in 1995 in Dublin, Ohio, is a 15-element metadata set similar to a library catalogue card and intended to aid discovery of electronic resources [7]. The 15 elements (covering resource content, intellectual property and instantiation) are: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage and rights.

The Unified Medical Language System (UMLS), which has been developed by the US National Library of Medicine since 1986 [8]. It is intended to help health professionals and researchers use biomedical information from different sources and is made by mapping many existing terminologies within a unifying framework. It comprises three knowledge bases: the Metathesaurus, a large repository of concepts, the Semantic Network, a limited network of 134 semantic types, and the Specialist Lexicon which corresponds to lexical resources.[8].

A short description of MedProAlg is given below:

The primary classes of MedProAlg are ProblemDescription and AlgorithmDescription. These are the classes in which expert users can describe the medical informatics problem and the algorithm that solves the problem respectively.

Every problem contains its static and its dynamic elements, which are classes named as StaticElement and DynamicElement respectively. The static elements of a problem are the ones that do not alter as time progresses, whereas the dynamic elements are changing in course of time.

The problem description (ProblemDescription class) consists of two slots, namely the problem_static_ description, and the problem_dynamic_description.

The static description (problem_static_description slot) of each problem is formulated when expert users enter the problem for the first time. The value type of the problem_static_description slot is instance and allows the StaticElement class, which in turn contains the following subclasses: CategorySchema class and Pragmatics class.

ISOFormat is a class containing: ISO31166, ISO 639-2, ISO8601 subclasses. The instances of these subclasses include the codes for the representation of names of countries, names of languages, dates and times encoded with the W3C encoding rules, respectively.

The dynamic description (problem_dynamic_ description slot) of each problem refers to the suggested algorithms (suggested_algorithm) and to various statistics (statistic_element).

The algorithm description (AlgorithmDescription class) also includes dynamic and static elements (algorithm_static_description slot, algorithm_dynamic_description slot), as well as the entry variables of the algorithm (algorithm_entry slot) and the specific problem that solves (reference_problem slot).

The value type of the algorithm_static_description slot is instance and allows the MAe:StaticElement class. MAe:StaticElement class in turn includes two slots: MAe:category and pragmatics. The value types of MAe:category slot and the pragmatics slot are instances that allow the MAe: CategorySchema and the Pragmatics classes respectively.

The value type of the algorithm_dynamic_description slot is instance and allows the MAe:DynamicElement class. MAe:DynamicElement class contains three subclasses, namely the UserMark, the When_Where_Be_ Used and the Automate_System_Mark.

The value type of the algorithm_entry slot is instance and allows the MAe:Variable class. The MAe:Variable class includes the following slots: description, measurement, name, variable_type, variable_name.

The value type of the reference_problem slot is instance and allows the ProblemDescription class.

An extensive description of MedProAlg Ontology is given in paper A Framework to describe problems and algorithms via ontology[9].

3. Scenario and architecture of the system

For the achievement of the goals set, a scenario concerning the creation of a MIPs and their respective algorithms knowledge base were employed. This knowledge base is constituted of the descriptions given by medical expert and computer scientists through MedProAlg Ontology. Moreover, a system architecture consisting of a knowledge base, web pages describing MIPs and their respective algorithms, and a frame of implemented algorithms is suggested.

As one can notice from the MedProAlg system architecture (figure 2) the scenario incorporates the following actions:

1. The system administrator builds the ontologies needed to describe the MIPs and the algorithms that solve them in RDF through Protégé – II. The ontology built for that cause is the MedProAlg ontology.
2. The MedProAlg ontology enters to MedProAlg application. This particular application is the one that automatically creates RDF descriptions of the MIPs and their respective algorithms, web pages derived from these descriptions, and finally the program framework, which will implement algorithms. It should be stressed at this point that RDF descriptions are based upon the structure framework resulting from the RDF/S of MedProAlg Ontology.
3. Medical Domain Experts and Informatics Domain Experts describe MIPs and their respective algorithms

using the Graphical User Interface of MedProAlg application. Furthermore, Informatics Domain Experts are responsible for completing the code of the algorithm framework, which is automatically generated from the descriptions and its final implementation.

4. Finally, medical researchers will be able to search the appropriate algorithm, which will solve a particular MIP. This MIP or a similar one will be the Knowledge Base of MIPs and their respective algorithms. Medical researchers will also be able to download and validate the implemented algorithm. Return

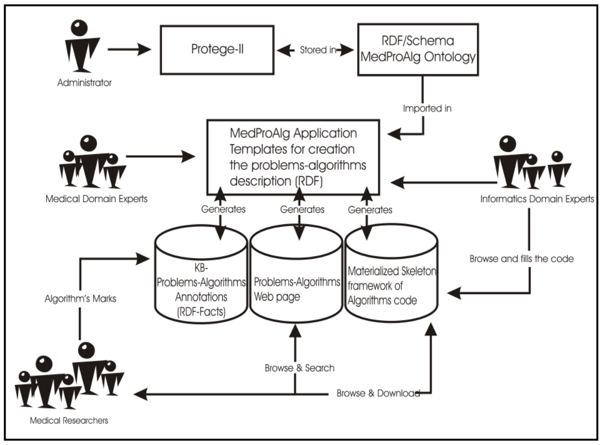

Administrator
Protege-II
Stored in
RDF/Schema MedProAlg Ontology
Imported in
MedProAlg Application Templates for creation the problems-algorithms description (RDF)
Medical Domain Experts
Informatics Domain Experts
Generates
Generates
Generates
KB-Problems-Algorithms Annotations (RDF-Facts)
Problems-Algorithms Web page
Materialized Skeleton framework of Algorithms code
Browse and fills the code
Algorithm's Marks
Browse & Search
Browse & Download
Medical Researchers

Figure 1: Architecture of the system

At this point, it is worth mentioning that during the MIP description (Action 3) and the algorithm search that solves a specific MIP (action 4), the doctor expert or the practitioner enter to the MedProAlg application two concepts which are the MIP category and the relevant keywords. The description of these concepts is automated. In particular, for the MIP category and keywords' descriptions, MedProAlg application is connected to UMLSKS (UMLS Knowledge Server). Then, each description is sent back as Concept Unique Identifier (CUI), a definition as documentation, broader relationships, narrower relationships, semantic type and synonyms of each concept.

The services of the UMLSKS can be accessed using one of the following methods: through a web client using a standard browser, through a program written to use UMLSKS API, through a TCP/IP socket-based interface.

For the MedProAlg Application (figure 3) the second method was employed. The API issues RMI request through the Internet directly to RMI server using RMI protocol, The RMI Server receives the request against the database, and formulates and returns the results to client API in one of two forms. The client program can request the information be returned in XML form and may subsequently use the Object Model classes to interpret the data as a set of data-centric objects. The object form is a set of classes that can be incorporated into the client application that provide functions for directly manipulating the database request results from within a java program.

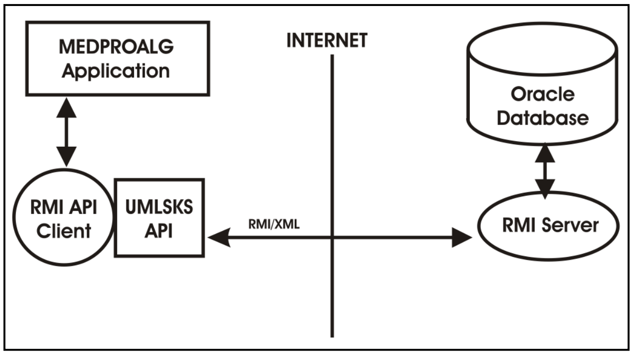


Figure 2: UMLSKS Connection.

4. Cardiology problems and algorithms

Numerous algorithms are available in Cardiology, and these may be used for diagnosis, management, monitoring or prognosis. Some typical algorithms include:

- Calculation of the body mass index (BMI)
- Calculation of anatomic areas and volumes from echocardiographic data
- Application of the Framingham data for predicting risk of cardiovascular disease
- Preoperative risk assessment prior to cardiac or noncardiac surgery.
- Identification of prognostic factors for short and long term prognosis following myocardial infarction
- Management of anticoagulation therapy in patients with atrial fibrillation
- Determining the severity of disease and level of disability

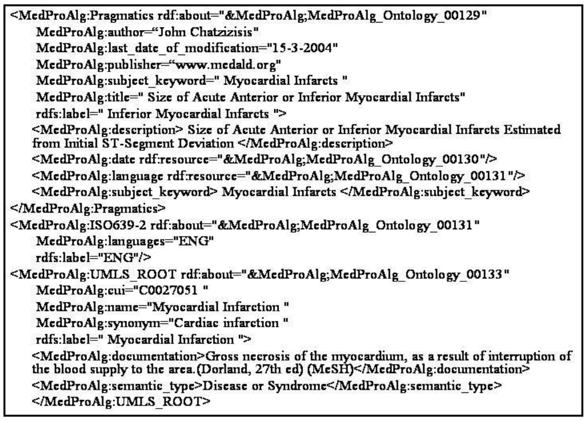
These algorithms may be used stand-alone, included in practice guidelines or embedded within medical devices. A web site www.medal.org currently makes available over 4,000 algorithms and includes about 200 cardiology-specific algorithms [3].

Thirty-two of the above mentioned cardiology algorithms are described by informatics experts and MICPs that they solve have been described by cardiology experts via MedProAlg Ontology.

For the MICP and algorithm description the MedProAlg Application was used. These descriptions resulted in the RDF of the MICP and the algorithms that solve them.

Figure 3 illustrates the description of one of the thirty-two MICPs in RDF named: ¨Size of Acute Anterior or

Inferior Myocardial Infarcts Estimated from Initial ST-Segment Deviation”



```
<MedProAlg:Pragmatics rdf:about="&MedProAlg;MedProAlg_Ontology_00129"
    MedProAlg:author=“John Chatzizisis"
    MedProAlg:last_date_of_modification="15-3-2004"
    MedProAlg:publisher=“www.medald.org"
    MedProAlg:subject_keyword=" Myocardial Infarcts "
    MedProAlg:title=" Size of Acute Anterior or Inferior Myocardial Infarcts"
    rdfs:label=" Inferior Myocardial Infarcts ">
   <MedProAlg:description> Size of Acute Anterior or Inferior Myocardial Infarcts Estimated
   from Initial ST-Segment Deviation </MedProAlg:description>
   <MedProAlg:date rdf:resource="&MedProAlg;MedProAlg_Ontology_00130"/>
   <MedProAlg:language rdf:resource="&MedProAlg;MedProAlg_Ontology_00131"/>
   <MedProAlg:subject_keyword> Myocardial Infarcts </MedProAlg:subject_keyword>
</MedProAlg:Pragmatics>
<MedProAlg:ISO639-2 rdf:about="&MedProAlg;MedProAlg_Ontology_00131"
    MedProAlg:languages="ENG"
    rdfs:label="ENG"/>
<MedProAlg:UMLS_ROOT rdf:about="&MedProAlg;MedProAlg_Ontology_00133"
    MedProAlg:cui="C0027051 "
    MedProAlg:name="Myocardial Infarction "
    MedProAlg:synonym="Cardiac infarction "
    rdfs:label=" Myocardial Infarction ">
   <MedProAlg:documentation>Gross necrosis of the myocardium, as a result of interruption of
   the blood supply to the area.(Dorland, 27th ed) (MeSH)</MedProAlg:documentation>
   <MedProAlg:semantic_type>Disease or Syndrome</MedProAlg:semantic_type>
   </MedProAlg:UMLS_ROOT>
```

Figure 3. Cardiology problem description in RDF format

5. Discussion and conclusions

Cardiologists and other physicians are faced with an enormous amount of information to process. Utilising appropriate algorithms can assist in processing this kind of information in a more effective way. In order to find out the most suitable algorithm for a particular problem it is necessary to know in detail the problem and all of its parameters. Given that problems concern medical science, whereas the algorithms that solve them are probably related to medical informatics, the creation of a problem and algorithm knowledge base via ontologies was recommended.

Medical algorithms or their description can be found within the web or in documents; however in most of the cases they are not structured and related to the problems that are able to solve. By the use of ontologies the prototype ontology MedProAlg ontology was built. This particular ontology is a structured framework able to describe MICPs and algorithms.

Thirty-two MICPs and their related algorithms were described via MedProAlg Ontology. It should be noted that a MICP can be solved by more than one algorithm.

The descriptions of those MICPs and their related algorithms create the MICPs and algorithms knowledge base. The architecture utilized to build this knowledge base helps to accomplish a manageable and efficient search of the most appropriate algorithms able to solve particular MICPs.

Furthermore, the description of each algorithm constructs automatically the framework of the program that implements it, through the proposed architecture. It is obvious that this process facilitates the programmer in implementing the algorithm.

The implemented algorithms will be available to doctors, who, in turn, will be able to validate them according to their user friendliness and their effectiveness when solving the problem. The final validation of an algorithm is an issue under study.

Finally, the use of UMLS, which standardizes the medical terminology, places this application in conformity with international standards.

Clearly, and due to its complexity, all module have aspects that may be improved:

- The implementation of MedProAlg application in order for the web pages to be automatically created.
- The implementation of MedProAlg application including the program skeleton automated creation.
- The improvement of MedProAlg ontology
- The description of further problems and algorithms
- The creation of an algorithms search engine.

References

[1] Iezzoni LI. The risks of risk adjustment. JAMA. e1997; 278: 1600-1607.

[2] McGinn TG, Guyatt GH, Wyer PC, Naylor CD, Stiell IG, and Richardson WS. Users' guides to the medical literature. XXII: Howto use articles about clinical decision rules. JAMA. 2000; 284: 79-84.

[3] Svirbely J, Sriram MG. MEDAL, a compendium of medical algorithms for access over the Internet. Proc AMIA Symp. 1999;p1172.

[4] Davies J, Fensel D, Harmelen F, Towards the semantic web, Ontology-driver Knowledge Management , Wiley &Sons Ltd, 2003.

[5] Merriam-Webster’s Collegiate Dictionary, http://www.m-w.com/dictionary.htm.

[6] Protégé-2000 an ontology tool, http://protege.stanford.edu

[7] Dublin Core Element , http:// www.dublincore.org

[8] Achour Ls, Dojat M, Rieux C, Bierling P, and Lepage E. A UMLS-based Knowledge Acquisition Tool for Rule-based Clinical decision Support System Development, J Am Med Inform Assoc 2001; 68(1):73-85.

[9] Ch Bratsas, N Maglaveras, P Quaresma, A Framework to describe problems and algorithms in medical informatics via ontologies, 4th European Symposium on Biomedical Engineering, Patra 2004.

Acknowledgements

This work was supported by PROMESIP and IRAKLITOS both funded by the Greek Ministry of Education (EPEAK I-II).

Address for correspondence.

Charalampos Bratsas
Aristotle University of Thessaloniki
Lab of Medical Informatics, POB 323
Thessaloniki, 54124 Greece
E-mail: mpampis@med.auth.gr