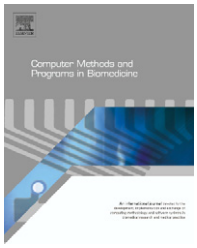
ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



KnowBaSICS-M: An ontology-based system for semantic management of medical problems and computerised algorithmic solutions

Charalampos Bratsas[a], *Vassilis Koutkias*[a], *Evangelos Kaimakamis*[a], *Panagiotis D. Bamidis*[a,*], *George I. Pangalos*[b], *Nicos Maglaveras*[a]

[a] *Lab of Medical Informatics, Faculty of Medicine, Aristotle University of Thessaloniki, P.O. Box 323, Thessaloniki 54124, Greece*
[b] *Informatics Laboratory, Computers Division, Faculty of Technology, Aristotle University of Thessaloniki, Greece*

ARTICLE INFO

Article history:
Received 26 March 2007
Received in revised form
29 May 2007
Accepted 27 June 2007

Keywords:
Medical Computational Problems
Semantic Description
Knowledge-based system
Ontology-based information retrieval
VSM-based Semantic Similarity
UMLS

ABSTRACT

In this paper, an ontology-based system (KnowBaSICS-M) is presented for the semantic management of Medical Computational Problems (MCPs), i.e., medical problems and computerised algorithmic solutions. The system provides an open environment, which: (1) allows clinicians and researchers to retrieve potential algorithmic solutions pertinent to a medical problem and (2) enables incorporation of new MCPs into its underlying Knowledge Base (KB). KnowBaSICS-M is a modular system for MCP acquisition and discovery that relies on an innovative ontology-based model incorporating concepts from the Unified Medical Language System (UMLS). Information retrieval (IR) is based on an ontology-based Vector Space Model (VSM) that estimates the similarity among user-defined MCP search criteria and registered MCP solutions in the KB. The results of a preliminary evaluation and specific examples of use are presented to illustrate the benefits of the system. KnowBaSICS-M constitutes an approach towards the construction of an integrated and manageable MCP repository for the biomedical research community.

1. Introduction

The value of algorithms and algorithmic processes in healthcare becomes obvious when considering the potential in enhancing clinical judgment by either validated clinical decision rules or quantitative methods [1]. A medical algorithm involves medical procedures encoding, data, information and knowledge in order to solve a clinical problem. Diagnostic Problem Solving (DPS) describes medical algorithms that refer to medical processes and are similar to medical prescriptions [2]. DPS applies to clinical reasoning strategies or clinical guidelines [3] and has recently witnessed many efforts to introduce standardised and classification approaches for representation and sharing [4,5]. On the other hand, Medical Computational Problem (MCP) solving relates to medical problems and their computerised algorithmic solutions. These solutions deal with mathematical or statistical models, related to data mining, signal or image processing, as well as, parameter estimation [6], aiming to enhance healthcare quality by supporting diagnosis and treatment automation. As more than 100,000 algorithms are already published in the literature, while some 8000 of them are already computerised, a new need, associated with the current poor organisation and questionable availability of them, arises [7]. These algorithms may

* *Corresponding author*. Tel.: +30 2310 999310; fax: +30 2310 999263.
E-mail addresses: mpampis@med.auth.gr (C. Bratsas), bikout@med.auth.gr (V. Koutkias), vkaimak@med.auth.gr (E. Kaimakamis), bamidis@med.auth.gr (P.D. Bamidis), pangalos@gen.auth.gr (G.I. Pangalos), nicmag@med.auth.gr (N. Maglaveras).
0169-2607/$ – see front matter © 2007 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2007.06.005

be used as either stand-alone, included in practice guidelines, or embedded within medical devices.

Currently, there is a plethora of medical algorithms available in the Web, which refer to MCPs and provide all computational facilities required to solve a medical problem. The main concern with MCPs is that the existing information about them is scattered and poorly organised. As a consequence, although general purpose search engines are fast in query answering, they do not focus on algorithmic solutions, and especially they do not provide results in a structured and comprehensive way. Typically, when searching for a particular MCP via a search engine, a huge result list is obtained containing links to resources that are not explicitly related to MCP content. Thus, too much time must be spent on browsing the results and very often the desired algorithm is either not found or the obtained description is not adequate.

Two specific efforts made in view of organising such information are PhysioNet and MedAl. PhysioNet (http://www.physionet.org/) is a Web-based resource supplying well-characterised physiologic signals and related open-source software to the biomedical research community. It was inaugurated in 1999 under the auspices of the US National Institute of Health (NIH) [8]. The Medical Algorithm Project (http://www.medal.org/) is another Web-based resource that disseminated medical algorithms in computer-executable forms, the vast majority of which were hitherto available only in paper-based media [9].

Although efforts such as the above-mentioned ones constitute significant repositories of MCPs, they lack semantic level qualities like comprehensive organisation and description. The primary aim of this work is the construction of an open and semantically enriched environment, providing the means for organising and managing unstructured/semi-structured and widely scattered information related to MCPs. To achieve the above purpose, *KnowBaSICS-M* (Knowledge-Based System for Integrating Computational Semantics in Medicine) was developed. The semantics of KnowBaSICS-M are defined in the *MCP Ontology*, an appropriate domain ontology that describes and classifies MCPs, in terms of problems description and their associated algorithmic solutions and implementations, providing the schema for the construction of a Knowledge Base (KB) for efficient use of MCP solutions. The *MCP Ontology* incorporates concepts from the Unified Medical Language System (UMLS) [10], in order to describe the medical terms via a controlled vocabulary. KnowBaSICS-M search mechanism relies on formal ontology-based queries in conjunction with an ontology-based *Vector Space Model* (VSM) [11,12], as an adjustment of the classic VSM [13], since ontology-based queries refer to purely Boolean Information Retrieval (IR) models that make sense when an information corpus can be fully represented as an ontology-driven KB, and do not provide clear ranking criteria [14,15].

KnowBaSICS-M aims to support clinicians, students or researchers in the fields of clinical medicine and medical informatics aiming to: (1) search for potential algorithmic solutions of a medical problem, (2) acquire knowledge about the specifications of algorithms, their implementation details and running environment, potential bibliographic resources, as well as download information about the relevant software whenever available, and (3) provide knowledge describing the semantics of new MCPs, constituting in this way an open framework for MCP research and management. However, to fully define an MCP, the contribution of practitioners or other medical scientists is required. For the solution of an MCP other fields of knowledge, such as statistics, mathematics, informatics, physics, etc., are also considered as prerequisites.

In the rest of the paper, the architecture of KnowBaSICS-M is presented, emphasising on the development of the *MCP Ontology* and the ontology-based IR technique employed. A functional scenario illustrating the interaction among the system's modules that correspond to MCP search and insertion follows. An experimental evaluation of the system's effectiveness is then presented, in terms of users' appreciation and estimation of precision/recall measurements. Two specific examples of the proposed system use are presented highlighting its medical impact and functionality. Finally, a discussion on our findings, a comparison with related works, as well as future work directions, conclude the paper.

2. Materials and methods

2.1. System architecture

KnowBaSICS-M follows a modular design comprising of four major subsystems as depicted in Fig. 1, namely, the *Semantic MCP Repository*, the *Medical Terms Annotator*, the *Query Engine* and the *Ontology Vector Space Model* (VSM). Interaction with KnowBaSICS-M is provided via an appropriate *User Interface* that encapsulates the functionalities for MCP search, retrieval and insertion. For each one of the aforementioned subsystems, a description follows in terms of its incorporated modules and functionality.

2.1.1. Semantic MCP repository

The *Semantic MCP Repository* constitutes the backbone of KnowBaSICS-M that conceptualises and manages the MCPs by means of defining, organising and structuring the associated knowledge model. It contains: (i) the *MCP Ontology*, (ii) the corresponding *MCP Knowledge Base* (MCP KB) and (iii) the *Knowledge Insertion Module* that acquires knowledge in the MCP KB (Fig. 1), as described below.

2.1.1.1. MCP ontology. It incorporates the conceptualisation of the domain knowledge for defining the MCP descriptions in the context of KnowBaSICS-M. It was constructed as an OWL (Ontology Web Language) ontology model [16]. In the context of this work, an MCP is formally represented as a triplet $(p, a, i) \in P \times A \times I$, where P is the medical problem space, A is the algorithm space and I is the implementation space, while p, a and i constitute instances of the triple space.

Fig. 2 illustrates the basic classes of the *MCP Ontology* along with their relations. In particular, in order to describe the medical problems' semantics in the MCP space, the *MedicalProblem* class was defined that is linked via appropriate attributes with the following classes:

- *Algorithm*: Corresponds to the medical problem solutions.
- *Profile*: Via its *MedicalProblemProfile* subclass (not shown in Fig. 2) it contains the medical problem description, the

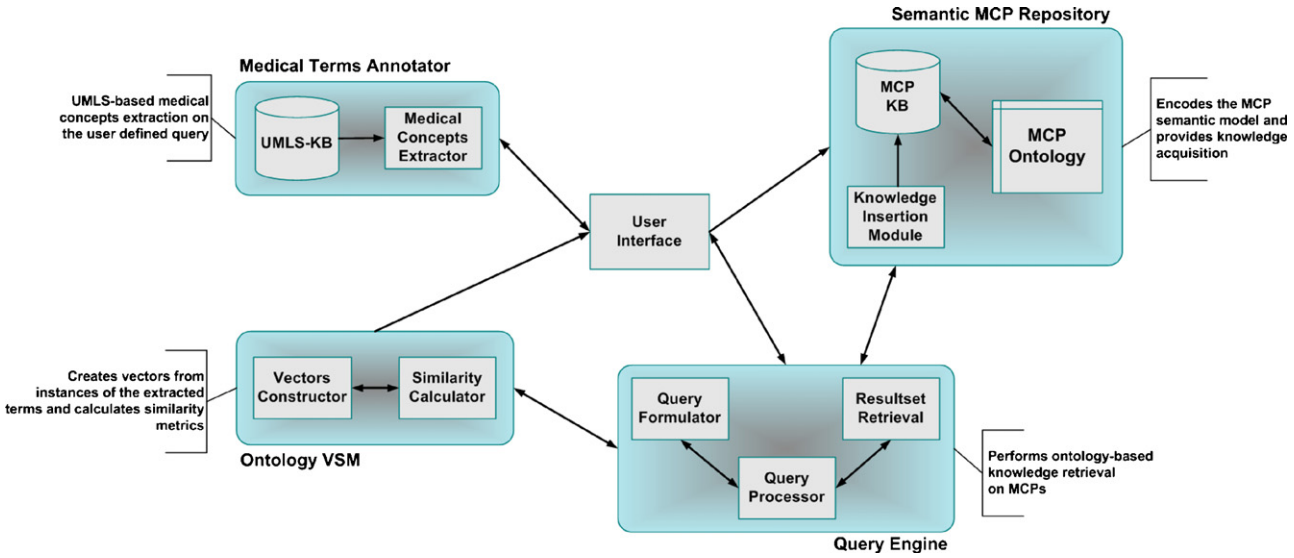


Fig. 1 – KnowBaSICS-M system architecture.

described language in ISO639-2 format (like EN, etc.), associated bibliographic references and potential modifications.

- *IndexTerm*: Includes keywords and their weights used in the *Ontology VSM* described in Section 2.1.4. Keyword descriptions follow the UMLS vocabulary (by linking to the *UMLS_Root* class).
- *Category*: Defines the medical problem category, the description of which follows the UMLS vocabulary (by linking to the *UMLS_Root* class).

The class describing the algorithms' semantics is the *Algorithm* class linked via appropriate attributes with the classes:

- *MedicalProblem*: The problem solved by the algorithm.
- *Profile*: Via its *AlgorithmProfile* subclass (not shown in Fig. 2) it contains the algorithm description.
- *AlgorithmModel*: Encapsulates the semantics of input and output variable information and the algorithm pseudo code. The type of inputs and outputs for each algorithm model

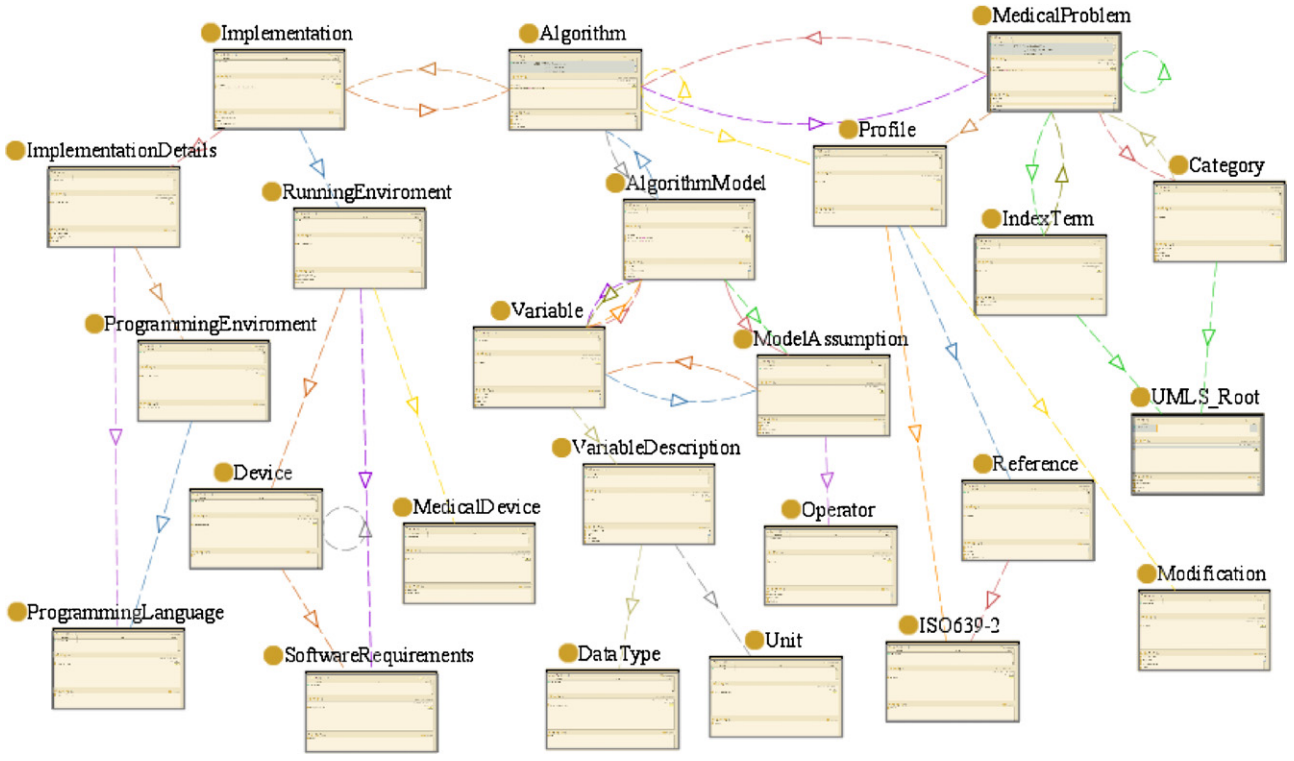


Fig. 2 – Part of the MCP Ontology depicting its major classes, i.e., *Algorithm*, *MedicalProblem* and *Implementation*, along with their relations with other classes of the knowledge model.

is defined by the *Variable* class. The *Variable* class is linked with the *VariableDescription* class, which includes the data types (*DataType* class) and the units (*Unit* class) for each input/output parameter of the algorithms.
- *ModelAssumption*: Defines a set of assumptions about each algorithm model, like pre-conditions of use, which impose requirements about the input data, and post-conditions, setting up requirements about output data. The *ModelAssumption* class is linked with the *Variable* and *Operator* classes (the later contains two subclasses, i.e., the *LogicalOperator* and the *ComparisonOperator*), in order to describe the algorithm model assumptions in a structure way.

Finally, the class incorporating semantics on algorithm implementations is the *Implementation* class linked via appropriate attributes with the classes:

- *ImplementationDetails*: Refers to the programming language and the programming environment that is used to implement the algorithm (e.g., Java as the programming language and Eclipse as the programming environment).
- *RunningEnvironment*: Refers to devices – medical or not – (i.e., hardware requirements) and to software requirements (e.g., execution environment, operating system, etc.).
- *Algorithm*: Corresponds to the implemented algorithm.

Moreover, in order to describe the users of KnowBaSICS-M, the *Users* class was defined in the *MCP Ontology*, which contains two subclasses, namely, the *SimpleUser* and *KnowledgeAuthor*. A knowledge author can search, browse and describe the semantics of the *MCP Ontology*, while simple users can only browse and search the MCP repository.

Aiming to describe the MCPs' bibliographic references in a structured manner, the *BibTex* OWL Ontology (available at: http://visus.mit.edu/bibtex/0.1) was adjusted and incorporated in the *MCP Ontology*. Moreover, the *ConOnto* software and hardware ontologies (both available at: http://www.site.uottawa.ca/~khedr/Ontologies/) were adjusted into the *MCP Ontology*, allowing for descriptions on software and hardware implementation profiles. The *Global Medical Device Nomenclature* (*GMDN*) relational schema (http://www.gmdn.org/GMDN_Technical_2003v2.pdf) was adjusted into the *MCP Ontology*, in order to formally describe potential medical devices associated with MCPs.

2.1.1.2. MCP Knowledge base. In the KnowBaSICS-M context, MCP-related knowledge is acquired according to the *MCP Ontology* that specifies its structure (entity types and relationships) and its classification scheme. The *MCP Ontology*, together with a set of instances of its classes, constitutes the *MCP Knowledge Base* (MCP KB). Medical problems stored as individuals in the MCP KB are represented in the *P* space as $p(x_1, x_2, \ldots, x_n)$, where x_i are instances of the related classes that describe the medical problem. Similarly, algorithms and implementations are stored as individuals in the MCP KB, corresponding to *A* and *I* spaces, respectively.

2.1.1.3. The knowledge insertion module. It is responsible for insertion/update of the MCP KB by creating and updating the instances of the *MCP Ontology*.

2.1.2. Medical Terms Annotator

The *Medical Terms Annotator* is used to annotate query MCP descriptions via terms obtained from UMLS. It consists of: (i) the *UMLS Knowledge Base* and (ii) the *Medical Concepts Extractor* (Fig. 1), as described below:

2.1.2.1. UMLS Knowledge Base (UMLS-KB). UMLS constitutes a medical lexical knowledge source along with a set of associated lexical programs [17]. It is used for terminology research, mapping between other terminologies, information indexing retrieval and Natural Language Processing (NLP). The knowledge source consists of the UMLS Metathesaurus, the UMLS Semantic Network and the SPECIALIST Lexicon [18]. In this work, the UMLS Knowledge Source Server (UMLS-KS) is used to access the UMLS Metathesaurus, while all vocabularies in the Metathesaurus are also used (i.e., MeSH, ICD9/10, SNOMED CT, etc.).

2.1.2.2. Medical Concepts Extractor (MCE). Maps information from the user-defined MCP query to UMLS concepts. In particular, the MCE applies basic NLP techniques by removing the common words (the stop words table is available at http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html#Stopwords) from the query, and matches the remaining phrase/word to UMLS concepts with normalised string-match criteria. Optionally, the MCE can restrict the extracted concepts by applying a user-defined semantic filter which removes the concepts of the semantic groups [19] that the user is not interested in. The MCE returns to the user a list of concepts along with their UMLS description, i.e., *Concept Unique Identifier* (*CUI*), *Definition, Semantic Type, Synonyms*.

2.1.3. Query engine

The *Query Engine* is aware of the *MCP Ontology* schema and consists of: (i) the *Query Formulator*, (ii) the *Query Processor* and (iii) the *Resultset Retrieval* module (Fig. 1). It encapsulates a mechanism *f* that maps user-defined queries Q into the MCP semantic space $P \times A \times I$, i.e., f:Q $\rightarrow P \times A \times I$.

2.1.3.1. Query Formulator. It is used to express the search criteria of the users in a formal ontology-based query language suitable for the *MCP Ontology*, e.g., in SPARQL [20]. In a general case, the search criteria could be based on any part of the *MCP Ontology* hierarchy related to MCP description, i.e., problem description, algorithmic solutions, specific algorithm inputs or/and outputs, model assumptions (e.g., preconditions), implementation details, etc. In the current implementation, we calculate the semantic similarity in the MCP corpus based on the problem description criteria that are set by the user, rather than enabling query formulation based on other more specialised criteria.

2.1.3.2. Query Processor. Incorporates the methods to read an ontology-based query from the *Query Formulator*. These methods return a query object, which encapsulates a parsed query. The next step is to create an instance of the execution query, which represents a single result of a query.

2.1.3.3. *Resultset Retrieval*. Retrieves a set of instances constituting the results of a user-defined MCP query, executed by the *Query Processor*.

2.1.4. *Ontology VSM*

The *Ontology VSM* provides a semantic similarity calculation mechanism of MCPs inserted in the MCP KB upon user-defined queries. It consists of: (i) the *Vector Constructor* and (ii) the *Similarity Calculator* (Fig. 1), as described below.

2.1.4.1. *Vector constructor*. It defines the weight vectors of the ontology's MCPs and the query MCP. The weights are computed automatically by an adaptation of the classic VSM for ontology-based IR, specifically via the *tf-idf* algorithm [13], i.e., based on the frequency of occurrence of the instances of a keyword property I_i in the *IndexTerm* class for each MCP description j:

$$w_{i,j} = \frac{\text{freq}_{i,j}}{\max_y \text{freq}_{y,j}} \times \log\left(\frac{N}{n_{i,j}}\right), \quad (1)$$

where $\text{freq}_{i,j}$ is the number of occurrences of I_i in the MCP description j, $\max_y \text{freq}_{y,j}$ the frequency of the most repeated instance I_y in the MCP description j, N the total number of MCP descriptions available in the search space and $n_{i,j}$ is the number of MCPs descriptions annotated by instance I_i.

These weights are stored as instances of weight property in the *IndexTerm* class (Fig. 2). Instances of *IndexTerm* class include the keywords through which an MCP description has been annotated along with their weights. In particular, in the insert process first the new MCP description and the relevant keywords are inserted in the MCP KB and then the MCPs' weights are recalculated. Vectors' construction involves the following steps:

- The *Vector Constructor* receives a set of instances of the *MedicalProblem* and *IndexTerm* classes from the MCP KB after the query execution. The query results are a set of tuples that satisfy the query. These tuples are also instances of the MCP KB and constitute the variables used by the *Vector Constructor* to form the MCPs and query vectors.
- The *Vector Constructor* defines the MCP vectors as $P_j = (p_{1,j}, p_{2,j}, \ldots, p_{m,j})$, $\forall j \in \{1, \ldots, \ell\}$, where m is the number of instances of the *IndexTerm* class, ℓ the number of instances of *MedicalProblem* class that satisfy the query, and $p_{i,j}$ is equal to $w_{i,j}$, if such a query result exists, or 0 otherwise.
- The *Vector Constructor* defines the query vector as $Q = (q_1, q_2, \ldots, q_m)$, where q_i is equal to $w_{i,q}$, if instance I_i appears in some tuple of the query result, and 0 otherwise. Weights $w_{i,q}$ are calculated using Eq. (1).

2.1.4.2. *Similarity calculator*. Its purpose is to rank the MCPs returned from the *Vector Constructor* according to their estimated relevance to the query. Several approaches for calculating the similarity measures in VSMs have been proposed so far [13], the most common of which is the *cosine coefficient*, which was adopted in KnowBaSICS-M in order to infer the similarity between the MCPs (P_j) and the query vectors (Q):

$$\cos(P_j, Q) = \frac{P_j \cdot Q}{|P_j| \times |Q|} = \frac{\sum_{i=1}^{m} w_{i,j} \times w_{i,q}}{\sum_{i=1}^{m} w_{i,j}^2 \times \sum_{i=1}^{m} w_{i,q}^2}. \quad (2)$$

2.2. *System functionality*

The functionality of KnowBaSICS-M is illustrated in the UML sequence diagram provided in Fig. 3. The basic operations of KnowBaSICS-M are the description of MCPs semantics in the MCP KB and the quest of MCPs through the *Ontology VSM*.

A user may need to either insert knowledge in the MCP KB or perform a search at its existing information regarding medical problems, specific algorithms or implementations.

The MCP query is analysed by the MCE which sends a request to the UMLS-KB in order to find the UMLS descriptions of terms or phrases contained in the text of the MCP query. The extracted UMLS concepts are returned to the user as a list of terms by the MCE. The user chooses the ones

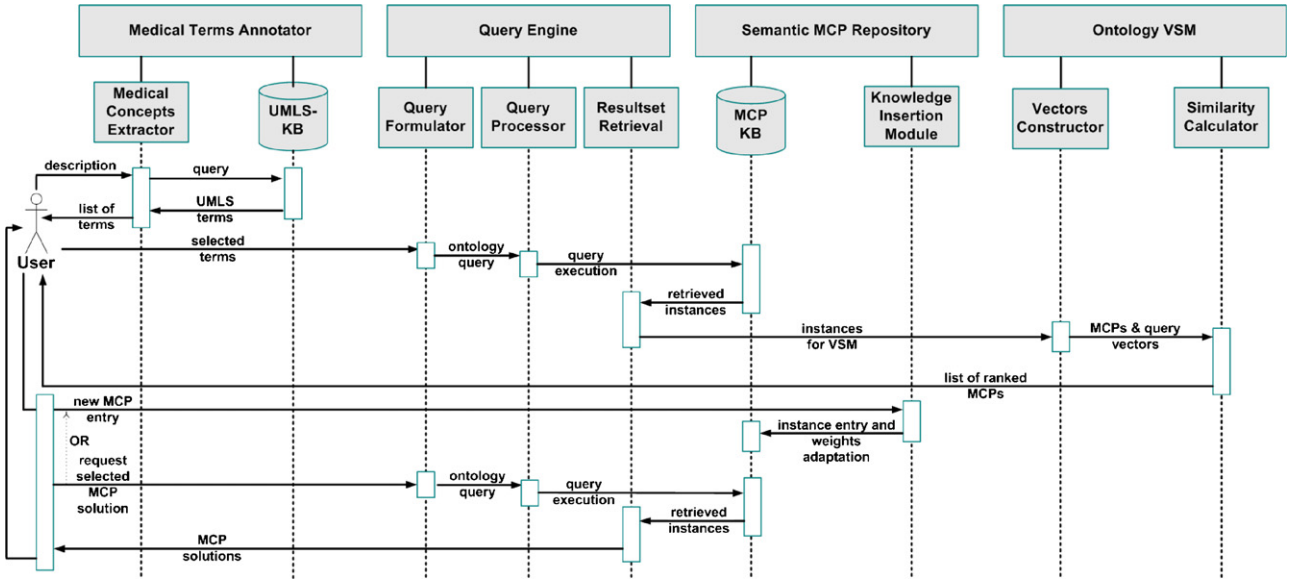


Fig. 3 – UML sequence diagram illustrating the procedures for MCP query execution and MCP insertion in KnowBaSICS-M.

upon which he/she wishes to construct the MCP keywords and these are sent to the *Query Engine*, which is aware of the *MCP Ontology* schema and initially creates a ontology-based query via the *Query Formulator*. The query is then executed in the MCP KB by the *Query Processor* and finally the *Resultset Retrieval* module receives the instances of the *MCP Ontology* that satisfy the query (if any) and forwards them to the *Ontology VSM*. Subsequently, the ontology-based MCP vectors and the quested MCP vector are created by the *Vector Constructor*. The *Similarity Calculator* receives these vectors and calculates the corresponding similarities. The MCPs with a similarity above a certain threshold are presented at the *User Interface* in a tabular, descending order form, containing the descriptions of the MCPs and the corresponding similarity scores. Furthermore, the user is able to browse the semantic description of each MCP (in terms of bibliographic references, algorithm solution specification, implementation specification, etc.).

In case the user determines that his/her MCP of interest is not identical as the highest ranked (or among the ones retrieved), the specific MCP may be inserted in the MCP KB. In that case, the *Knowledge Insertion* module receives the description of a new MCP and creates the new instances in the MCP KB by additionally adapting the weights of the MCPs keywords in the *MCP Ontology*, which are recalculated every time a new MCP is inserted in the MCP KB.

2.3. *Implementation*

A client–server based ontology management system was built, allowing users to interact with the *MCP Ontology* in a user-friendly manner. Code development was based on open-source development platforms and tools. Specifically, the *Protégé* knowledge modeling tool and particularly its OWL plug-in [21] was used to construct the *MCP Ontology*. Consistency checking of our ontology model was performed via the RacerPro reasoner [22], which was also used in order to classify the *MCP Ontology* and compute inferred types of individuals.

From an implementation viewpoint, the system consists of the following parts:

- *KnowBaSICS-M Server*: It incorporates all the subsystems of KnowBaSICS-M described. The *Semantic MCP Repository* uses Jena [23] to parse the *MCP Ontology*. It creates a correspondence of each ontology concept to a Java object, which can then be arbitrarily manipulated. The *Medical Terms Annotator* is based on the UMLS Application Programming Interface (API) [10] to connect to the UMLS-KS [24], which in turn offers an open interface for querying its medical terminology semantic network, i.e., the Metathesaurus, the Semantic Network and the SPECIALIST Lexicon. The *Query Engine* relies on SPARQL, an ontology-based query language that is supported by Jena via the ARQ-API [23]. Finally, the *Ontology VSM* subsystem is a custom Java implementation.
- *KnowBaSICS-M Client*: It constitutes a graphical user interface that enables access to the full set of functionalities offered by the KnowBaSICS-M Server, i.e., searching, browsing and populating the corresponding MCP KB. It connects to the KnowBaSICS-M Server through Java RMI (Remote Method Invocation).

3. Example scenarios

In this section, specific examples of KnowBasics-M usage are presented in order to demonstrate the functionality of the system, highlighting also its medical impact. In particular, we illustrate an example of MCP retrieval

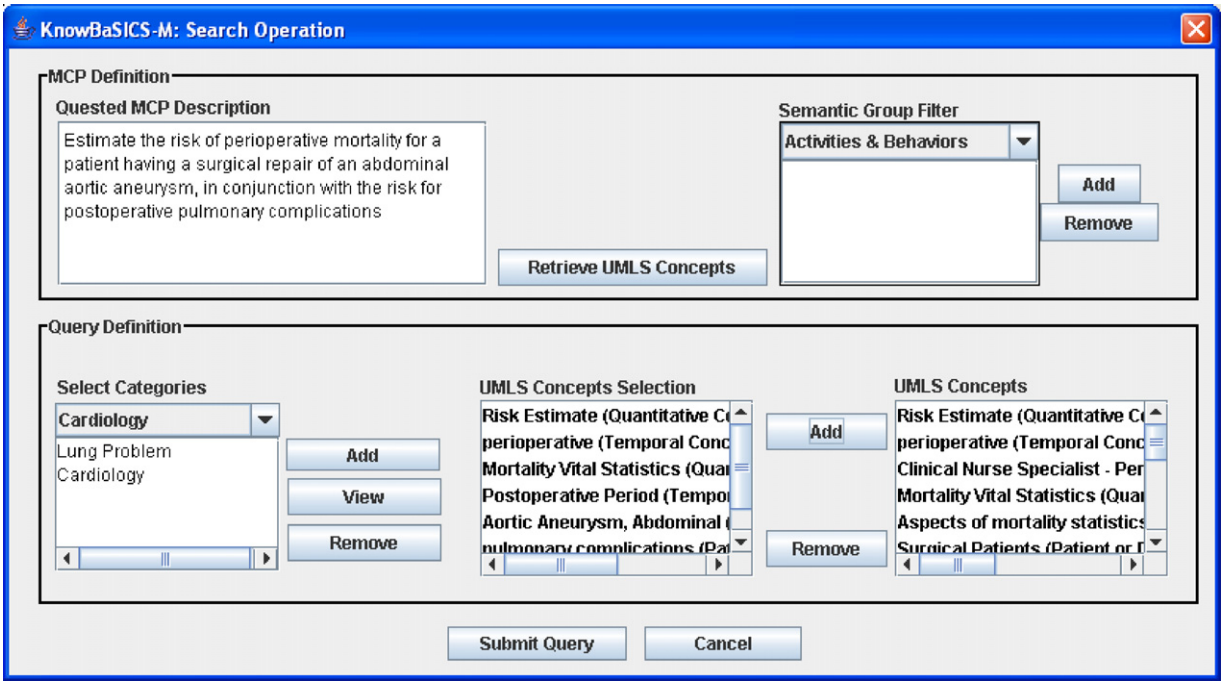


Fig. 4 – Query formulation example.

and another of MCP insertion via the KnowBaSICS-M user interface.

Example 1 *(Retrieve MCP)*. Estimate the risk of perioperative mortality for a patient having a surgical repair of an abdominal aortic aneurysm, in conjunction with the risk for postoperative pulmonary complications.

The first example illustrates use of the system by a physician evaluating a patient undergoing surgical repair of an abdominal aortic aneurysm, specifically for the risk of post-operative pulmonary complications. The user's goal is to identify surgical risks for the patient and to determine prognosis. The doctor sought an implemented algorithm or algorithms as a solution(s) to the described problem. He first provided, through the KnowBaSICS-M user interface, the description of his MCP of interest (Fig. 4). After query matching in the UMLS-KS, the user selected from the returned list of UMLS concepts the following terms (the corresponding UMLS CUIs are provided in parentheses): *Perioperative* (C1518988), *Postoperative* (C0032790), *Risk Estimate* (C0035647), *Mortality* (synonym to *Mortality Vital Statistics*, C0026565), *Surgical repair* (C0374711), *Abdominal Aorta Aneurysm* (C0162871) and *Pulmonary Complications* (C0281169). Finally, he selected the *Lung Problem* (C0740941) and *Cardiology* (C0007189) categories which the quested MCP belongs to.

Upon query submission, the *Query Engine* created and executed the following SPARQL query against the MCP KB based on the terms' CUIs:

```
SELECT ?MedicalProblem ?CUI ?Weight

WHERE { ?MedicalProblem :hasCategories ?Category.

?Category :hasCategoryDescription ?CategoryDescription.

?CategoryDescription :hasCui ?CategoryCUI.

?MedicalProblem :hasMedicalConcepts ?IndexTerm.

?IndexTerms :hasKeywordWeight ?Weight.

?IndexTerms :hasKeywordDescitpion ?UMLSDescription.

?UMLSDescription :hasCui ?CUI.

FILTER((?CategoryCUI ="C0740941" || ?CategoryCUI ="C0007189") &&

(?CUI="C0032790"||?CUI=" C0035647"||?CUI=" C0026565

"||?CUI="C0374711"||?CUI="C0162871"||?CUI =" C0281169"))}
```

The results obtained after executing the query were a set of the following *MCP Ontology* instances: the medical problems, the CUIs of those concepts that were set as keywords for the description of each medical problem and the respective keywords' weights. The *Ontology VSM* first constructed the MCPs vectors from these instances and then calculated their similarities.

As a consequence, a number of MCPs were obtained as results relevant to the clinical question after using the search process described above (Fig. 5). The first two of them were the solutions the doctor had been searching for, since they constitute the two aspects of the solution process, both leading to the overall estimation of the desired feature, i.e., the overall perioperative risk for that patient. The user could see the details of the algorithms or the implementations of the solving problem by performing a request for the selected MCP solutions (input/output data, pseudo code, running environment, references, etc.). Moreover, the user had the ability to download the possible implementations of the chosen algorithms and was also able to use a direct Web link to the published resources relevant to the algorithms of interest and thus exercise evidence-based medicine.

In comparison, the same search was conducted by the physician using the exact keywords at the Google search engine, as well as directly at the MedAl repository via its search functionality. In the first case, a huge number of results were obtained (e.g., more than 39,700 results in 0.25 s), which did not refer, however, to structured medical algorithms especially in an executable form, and the browsing/review of the results required much time to be dedicated by the user. In the MedAl search, on the other hand, not all the desired MCPs were returned, since the incorporated search mechanism tries to combine the MCP title with the given keywords using string exact match criteria without relying on semantic criteria. For example, if we input the terms "aneurysm of abdominal aorta" as keywords, MedAl does not return any results, while in KnowBaSICS-M, due to the conjunction of the *MCP Ontology* with UMLS (which provides a standard vocabulary for the searched keywords, i.e., phrases/words), the above-mentioned results are the same. Moreover, KnowBaSICS-M supports the combination of words in phrases, estimating their semantic similarity in the MCP KB through the *Ontology VSM*.

Example 2 *(Insert MCP)*. Clinical evaluation of a child from 1 to 5 years of age hospitalised for asthma using a severity score.

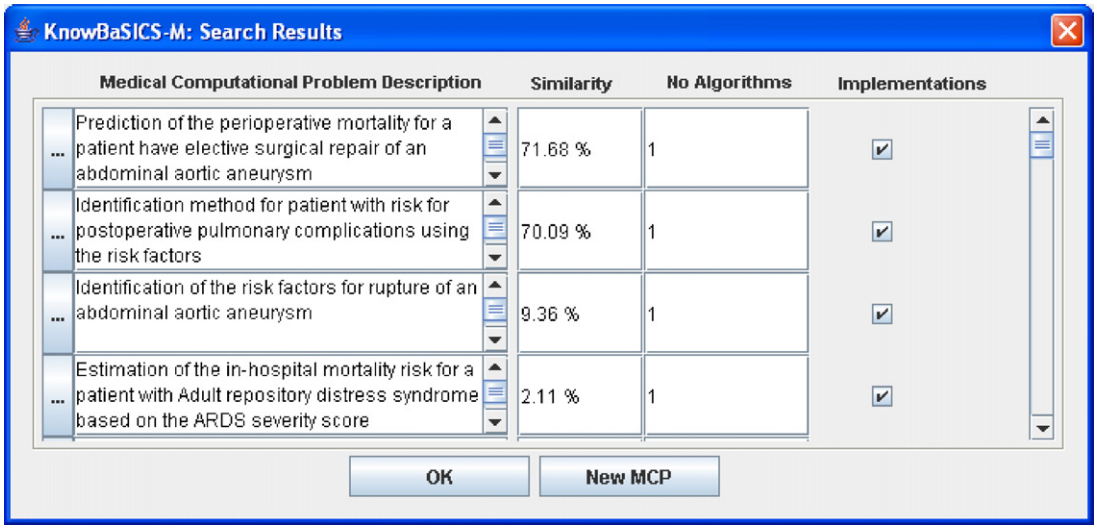


Fig. 5 – MCP retrieval example.

In this example, a user wanted to insert a new pulmonary-related MCP into KnowBaSICS-M. The user initially searched whether the pulmonary MCP existed already in the MCP KB. The clinical purpose was the estimation of the severity of asthma in a child from 1 to 5 years of age who had been hospitalised because of the disease. The hospitalisation parameter here was considered of high importance, dictating a different severity range and a different treatment strategy. By choosing as index terms of the query the next concepts: *Clinical evaluation*, *child*, *age–years*, *hospitalised*, *asthma* and *severity score*, the most similar MCP retrieved from MCP KB was: "*Clinical evaluation of the severity of clinical asthma in school age children using the severity score*" with a similarity of 73.87%. The user decided that the quested MCP was not included in the system and proceeded to its insertion (Fig. 6). During the insertion process, the descriptions of the new medical problem, as well as its index terms were automatically registered in the MCP KB, while the weights of keywords were also adapted. The author had the

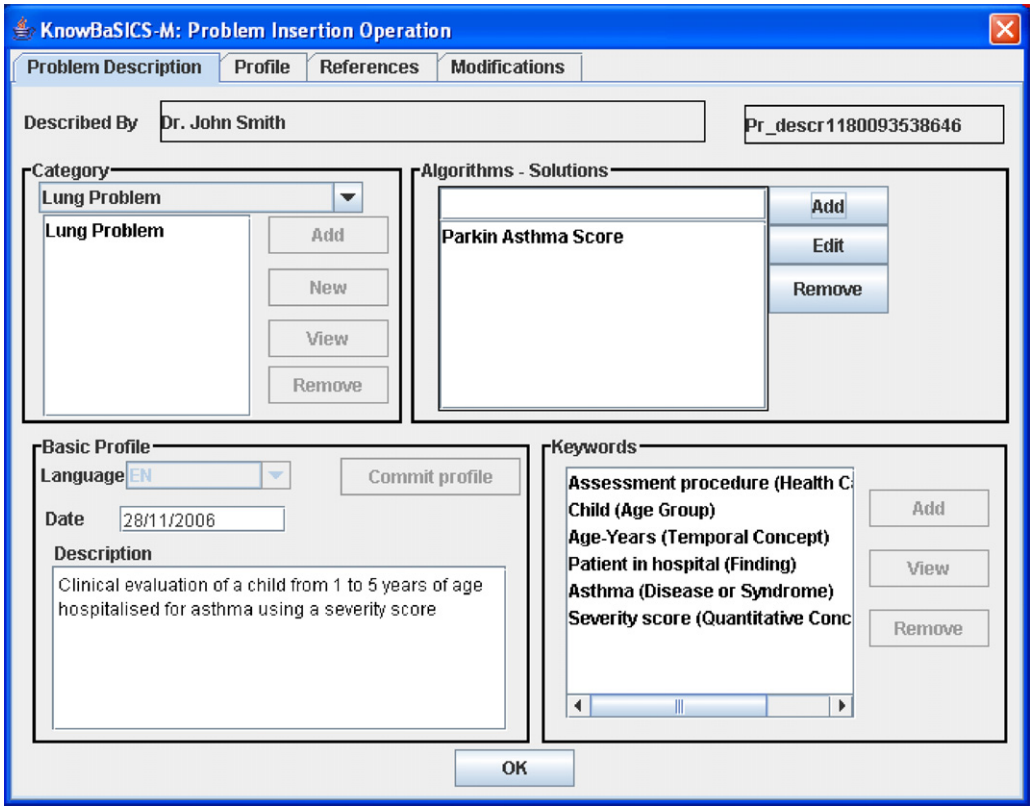


Fig. 6 – MCP insertion example.

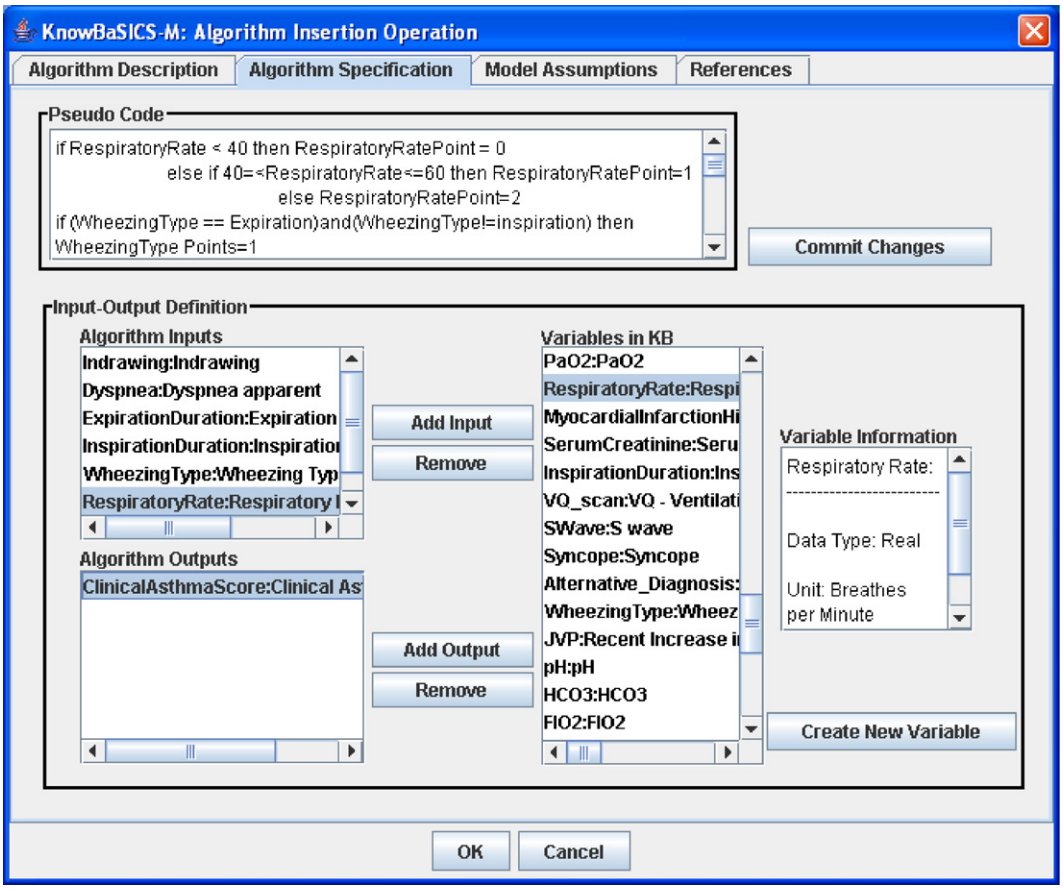


Fig. 7 – Algorithm semantic description example.

ability to fill-out possible references dealing with the problem and its potential future modifications.

Moreover, the author defined the semantics of the algorithm (Fig. 7), by selecting inputs/outputs from existing variables or by creating new ones in the MCP KB, providing the algorithm pseudo code, defining the algorithm possible assumptions and potentially providing relevant bibliographic references. Similarly, the author described the semantics of the implementation(s) of the algorithm (Fig. 8).

4. Experimental evaluation

In view of a preliminary experimental evaluation, KnowBaSICS-M was tested by physicians on a corpus of MCPs retrieved by the MedAl repository [9]. These algorithms formed the MCP KB. Specifically, two categories were examined, namely cardiology and pulmonary medicine, which included 123 and 109 MCPs, respectively. The MCP KB included in total 13,748 instances. For purposes of evaluating the accuracy of KnowBaSICS-M, the system was used by a number of physicians. A total of four users participated in the evaluation process and their remarks on using the system were recorded along with the obtained results. The first scope of the study was to evaluate KnowBaSICS-M either for knowledge insertion or for knowledge retrieval in order to assess its usability. A secondary aim was to calculate the precision and recall features.

The test physicians were familiar with the MedAl project and they were asked to form a number of clinical questions from the fields of cardiology and pulmonary medicine, encouraging them to express their queries in a very descriptive, natural language like form and without providing any other instructions or training concerning the keywords selection. The latter was done in purpose, since one of the aims of the evaluation study was to examine how the system corresponds to different descriptions according to the knowledge background of each medical specialist. A total of 68 clinical questions were addressed by the physicians. After defining the search criteria via KnowBaSICS-M, for all questions and before obtaining the search results and reviewing the returned MCPs, the users were given the full range of the existing MCPs in the MCP KB in the field of the specialty or specialties of interest. The physicians then performed the manual marking of the relevant MCPs residing in the MCP KB that corresponded to their clinical questions.

The comparison between the manual marking of the MCPs and the obtained results from the system revealed the following: Four questions were not answered at all from the existing MCP repository and the users defined them as new MCPs. The users found acceptable answers to their quest in 51 of the remaining 64 questions, while 13 questions were defined as new MCPs. The system returned MCPs with similarity between

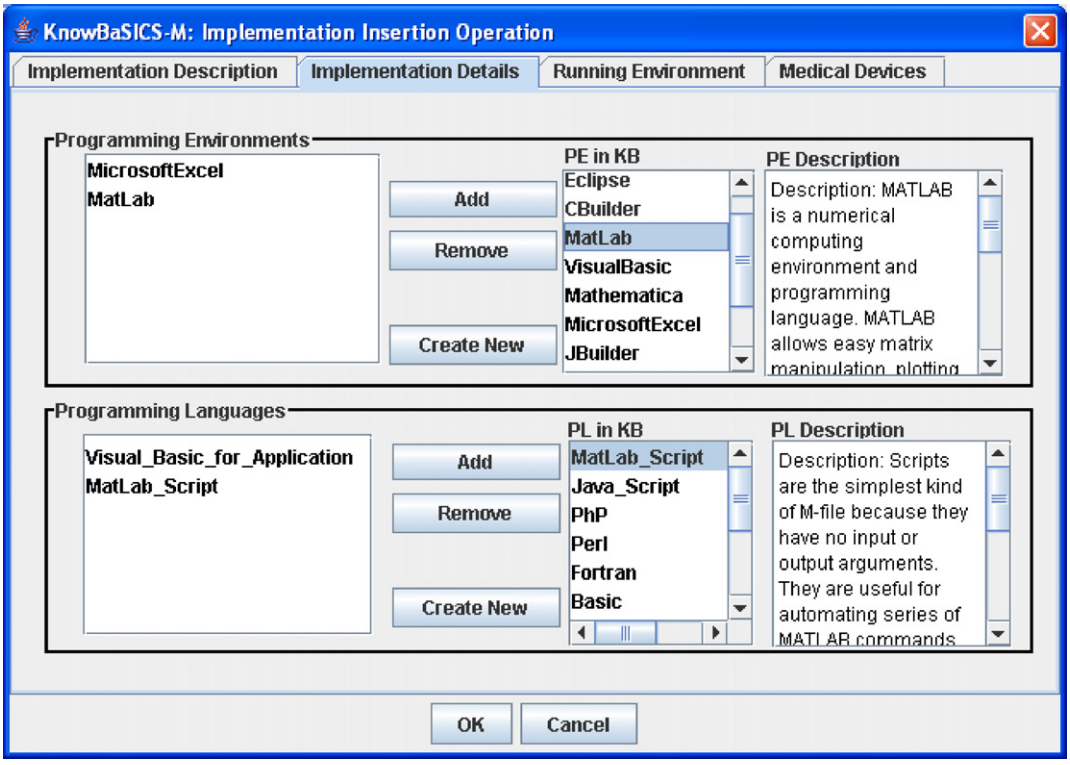


Fig. 8 – Implementation semantic description example.

61% and 74% in 5 out of these 13 questions but the users did not find the results satisfactory to their quests whilst the answers of the other 8 of the 13 questions had similarities less than 40%—unsatisfactory results according to the users too. In particular, 65 relevant MCPs were manually identified, 32 of which constituted the same medical problem as the quested MCP. After executing the search process, we found that 18 out of those 32 were returned with similarities over 80%, while the remaining 14 had similarities over 85%.

Aiming to calculate the *precision/recall* in relation to the similarity level estimation, the returned MCPs above each similarity level were considered as *true positive* when the physicians found their algorithms satisfactory for the solution of their quested MCP without expressing the necessity to insert a new medical problem – along with the corresponding solution – to the MCP KB. It is worth noticing that at similarity level over 70%, the average precision and recall characteristics for the users were approximately 87.36% and 89.84%, respectively. Although a specific similarity threshold was not used as a cut-off value in this case, due to the relatively small number of applied MCPs and questions, the above value is indicative of a proposed threshold for future use and remains to be proven after further evaluation of KnowBaSICS-M.

Furthermore, the users were quickly familiarised with the operation instructions given to them and were able to use the system effectively. The overall impression of the system usage was very satisfactory and the users reported that they found the application easy to interact with, effective, and very helpful in their quest for specific medical algorithmic solutions. The physicians appreciated the presentation of the similarity level next to the matching MCPs and they reported that the order of appearance of the returned MCPs was close to the actual similarity level to the original question. They also stated that such MCPs have a high occurrence in the everyday practice of clinical medicine and a system of this type could gravely help towards the direction of automating the evaluation/decision process and ensuring delivery of evidence-based medicine.

It is evident, that in the scope of this preliminary evaluation study, it is not possible to draw safe and accurate conclusions regarding more specific usability issues such as the time required for MCP insertion, which is variable to user's familiarity with the system and the notion of computerised medical algorithms, user's computer skills etc. However, the time recorded for MCP insertion in this study was approximately 2–4 min. Although this feature may be considered indicative, it is necessary to perform a wider scale evaluation study, in which more users will participate in and a larger MCP corpus of various categories will be employed.

5. Discussion

5.1. *Related work*

It has recently become a great research challenge to organise and annotate the plethora of biomedical resources at a semantic level. To this extent, UMLS offers a lexical resource (the *Special Lexicon*), a terminology resource (the *Metathesaurus*) and an ontological resource (the *Semantic Network*), all integrated for this purpose. Although UMLS is considered as one

of the most significant knowledge resources for the biomedical domain, it is not sufficient for MCPs description, where a more specific semantic model is required.

Closely related to MCPs, MedAl [9] and PhysioNet [8] constitute significant efforts toward the construction of centralised repositories of medical algorithms and their associated implementations. These repositories provide vast information about the algorithms provided, in an unstructured form though, lacking descriptions of problems and algorithms at a semantic level. Considering also the heterogeneity of the descriptions that each repository adopts, the construction of a semantically enriched and generalised MCP space seems a favorable approach. In this context, KnowBaSICS-M was conceptualised as a modular system aiming to provide organisation, retrieval and management of knowledge related to MCPs. Similar to the general objectives of KnowBaSICS-M are found in OpenCPS [25], where problems and algorithms are semantically described and made available via a Web portal. However, the underlying ontology model does not consider the biomedical domain, while the techniques employed for ontology-based IR are not described.

Currently, there are several approaches that employ ontology-based IR techniques reported in the literature, which do not target the medical domain though. For example, Song et al. [26] proposed an ontology-based IR model for the Semantic Web based on the *tf-idf* algorithm and presented a scenario of retrieval that reflects the essential difference between the classic VSM approach and ontology-based VSM. Castells et al. [11] proposed an ontology-based schema for document annotation. They also proposed an IR model based on an adaptation of the classic VSM approach, including annotation and ranking algorithms. Their experiments showed clear improvements with respect to keyword-based search. They used the *KIM Ontology*, an upper-level ontology suitable for open-domain and general-purpose semantic annotation, in order to annotate documents. In addition, an ontology-based IR system for the Semantic Web was proposed by Kohler et al. [27], which incorporated fully automated methods for mapping equivalent concepts of imported ontologies, such as WordNet, by combining an ontology-based indexing mechanism and concept-based IR.

In our approach, semi-automatic indexing concerns user-defined free-text queries regarding MCPs or the provision of new MCP descriptions that are based on the *MCP Ontology* and the controlled vocabulary of UMLS. Specifically, IR is achieved by an ontology-based VSM similar to the one presented in reference [11]. However, we use UMLS as a controlled vocabulary in order to index the MCP queries and the MCP descriptions as instances of the *MCP Ontology*, instead of indexing them via general annotations by using upper-level ontologies such as KIM. Moreover, we employed basic NLP techniques, in order to construct the semantic queries, instead of directly formulating the semantic query in terms of the *MCP Ontology*. Ontologies and NLP techniques for term indexing were also employed in reference [27], where indexing was based on comparisons of the "word to be indexed" context to that of a concept in the underlying ontology. In particular, an ontological indexing process was provided, in order to map the words in the text to ontology concepts, while in our approach, we first map the words/phrases contained in the user-defined queries to UMLS concepts and in the following the retrieved UMLS concepts to *MCP Ontology* concepts.

Lately, there is also great interest in ontology-based IR and extraction systems applied in the biomedical literature, such as TextPresso [28] and GoPubMed [29]. TextPresso uses an ontology comprising of 33 categories of biomedical terms and includes all terms of the Gene Ontology (GO) in order to retrieve and extract information about particular biological facts, such as gene–gene interactions, from publications related to *C. elegans*. GoPubMed uses GO in order to annotate abstracts that are available via PubMed. Specifically, it submits user-defined keywords to PubMed, extracts GO terms from the retrieved abstracts, and presents the induced ontology browsing and the corresponding annotated abstracts.

The abovementioned approaches adopt ontology-based IR models that are not sufficient for annotating and organising MCP-related information. Thus, a more specialised ontological schema is required for MCP description and management. Moreover, scientific papers regarding MCPs are often describing in detail the medical problem that authors are trying to solve, as well as the algorithmic solution methodology and the results obtained. The details of the algorithms or their implementations are often missing or they are insufficiently described. Consequently, an automated system that could retrieve MCP-related information from the literature and populate the MCP KB accordingly was not considered as a feasible approach. In addition, KnowBaSICS-M uses ontology-based IR techniques in order to query the MCP KB and extract MCPs' semantic information, thereby, allowing its users to assess the validity of the retrieved algorithm(s) by studying the references and the corresponding descriptions provided attached to each query result, as well as the execution details of the algorithms.

5.2. *Future work*

Nowadays, a lot of research is being conducted for the construction of automated or semi-automated mechanisms enabling extraction of medical concepts from a medical text or medical records. KnowBaSICS-M automatically extracts the medical concepts from user-defined queries via direct connection with UMLS-KS. However, the final selection of the indexing terms is made by the user. Other approaches provide fully automated indexing of medical documents based on UMLS [30,31] by using NLP tools such as MedLee (Medical Language Extraction and Encoding System), MetaMap [17] and Negex2 (a computational algorithm using regular expressions) [32]. The modular design of KnowBaSICS-M enables incorporation of such NLP tools in order to generate semantic queries based on the retrieved indexing terms. Such a potential is under consideration, requiring, however, more thorough evaluation upon adoption.

An interesting perspective and extension of KnowBaSICS-M functionality is the evaluation of the algorithms that are incorporated in the MCP KB. In this case, tailored evaluation forms, like discerns' criteria [33], could be of great value. It is possible to evaluate an algorithm with objective criteria which include the publication source (the *Reference* class of the *MCP Ontology* can provide this kind of information). The assess-

ment of an algorithm can also be performed using levels of validation similar to the levels of evidence in "evidence-based medicine", as discussed in reference [34]. A subjective element based on the users' evaluation may be also present. In a wider sense, incorporation of a quality assurance mechanism assessing the MCP content inserted in the system is currently being elaborated, since quality issues are of major importance in all cases of "open" environments, where digital media are organised and managed by community users.

Another future consideration is an expansion of the system that will enable it to discover possible relations or links between various medical algorithms and propose their use in terms of consecutive procedures, in order to construct advanced MCP workflows. Such an extension is supported by the *MCP Ontology* via the input, output and algorithm precondition parameters of use. Considering an advanced DPS environment this direction could facilitate the identification of gaps in algorithm development by looking at their distribution relative to the different steps and stages outlined.

From a pure technical viewpoint, we consider to elaborate more on the user interface design, e.g., providing a tree-like structure, generated from a part of the *MCP Ontology*, for formulating queries on the MCP corpus, and a wizard-like procedure for MCP insertion. Another major technical challenge is the automated incorporation of the content located at existing repositories such as MedAl in the MCP KB. However, this is associated with several difficulties. More specifically, besides using integration schemas such as wrapper-mediation based, which are indicated in cases of heterogeneous, autonomous and online resources, the most important issue is the construction of an effective mean to match the unstructured/semi-structured information retrieved from other repositories to the *MCP Ontology* structure. It is clear that a fully automated approach to cope with such an integration issue is very complicated, requiring cooperation with the relevant MCP repository providers.

Finally, taking into account a clinical setting, an extension of KnowBaSICS-M is considered to support the automated identification of individualised algorithms that will be linked with Electronic Health Record (EHR) data, identifying, for example, which algorithms are applicable for specific patients based on medical data availability and the preconditions of algorithms.

5.3. *Conclusions*

KnowBaSICS-M is an open and extendable knowledge-based system, aiming to semantically describe and organise currently available heterogeneous and unstructured/semi-structured MCP descriptions and their associated solutions. Our aim was not only to create an efficient means of extracting, organising and visualising scientific knowledge, but also to adopt a medical problem centred approach that focuses on encouraging collaborative efforts in organising and sharing knowledge. Semantic IR techniques based on the classic VSM were used in order to achieve an appropriate search process by constructing MCPs and quested vectors from the MCP KB and estimate the similarity among them. While our experimental evaluation focused on inserting knowledge or finding MCPs from an inaugural MCP KB, promising results were achieved with high precision and recall index values. Based on this MCP corpus, an experimental evaluation of the system was conducted for two categories of Medical Computational Problems, in which users were able to obtain integrated solutions to their MCPs along with additional information needed for their execution. It is our intention to further evaluate the system, including both a larger number of test users and a wider range of MCPs concerning additional medical specialties, resulting in a more reliable estimation of the precision and recall features, and a more comprehensive usability study of the system.

KnowBaSICS-M constitutes an approach towards the construction of an integrated and manageable MCP repository for the biomedical research community. Thus, thorough use of such a system is expected to enhance task automation, cost containment and quality services in medical care, while at the same time medical research and high quality medical education by means of focused problem-based learning are going to be benefited at a considerable rate.

Acknowledgement

This work was supported by IRAKLEITOS funded by the Greek Ministry of Education—Operational Program for Educational and Vocational Training II (EPEAK II).

REFERENCES

[1] T.G. McGinn, G.H. Guyatt, P.C. Wyer, C.D. Naylor, I.G. Stiell, W.S. Richardson, Users' guides to the medical literature. XXII: how to use articles about clinical decision rules, J. Am. Med. Assoc. 284 (2000) 79–84.
[2] D.S. Hartley, in: J. Fawcett, D.J. Stein, K.O. Jobson (Eds.), The Language of Algorithms, In Textbook of Treatment Algorithms in Psychopharmacology, John Wiley and Sons, New York, 1999, pp. 15–31.
[3] J.F. Arocha, D. Wang, V.L. Patel, Identifying reasoning strategies in medical decision making: a methodological guide, J. Biomed. Inform. 38 (2) (2005) 154–171.
[4] M. Peleg, et al., Comparing computer-interpretable guideline models: a case-study approach, J. Am. Med. Inform. Assoc. 10 (1) (2003) 52–68.
[5] P.A. de Clercq, J.A. Blom, H.H. Korsten, A. Hasman, Approaches for creating computer-interpretable guidelines that facilitate decision support, Artif. Intell. Med. 31 (1) (2004) 1–27.
[6] C. Bratsas, P. Quaresma, G. Pangalos, N. Maglaveras, Using ontologies to build a knowledge base of cardiology problems and algorithms, Proc. IEEE Comput. Cardiol. (2004) 609–612.
[7] J.R. Svirbely, M.S. Iyenga, Issues in the implementation of computer-based medical algorithms, Technol. Health Care 13 (5) (2005) 438–439.
[8] I.C. Henry, A.L. Goldberger, G.B. Moody, R.G. Mark, PhysioNet: an NIH research resource for physiologic datasets and open-source software, in: Proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems, 2001, pp. 245–250.
[9] G. Kantor, J. Svirbely, K. Johnson, M. Sriram, J.R. Rodriguez, J. Smith, MedAl: the medical algorithm project, in: Proceedings of MEDINFO, 2001, p. 298.
[10] Current UMLS Release—2007AA, available at: http://www.nlm.nih.gov/research/umls/documentation.html.

[11] P. Castells, M. Fernández, D. Vallet, An adaptation of the vector-space model for ontology-based information retrieval, IEEE Trans. Knowl. Data Eng. 19 (2) (2007) 261–272.
[12] C. Bratsas, I.S. Hatzizisis, P. Bamidis, P. Quaresma, N. Maglaveras, Similarity estimation among OWL descriptions of computational cardiology problems in a knowledge base, Proc. IEEE Comput. Cardiol. (2005) 243–246.
[13] G. Salton, M. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.
[14] P. Castells, et al., Neptuno: semantic web technologies for a digital newspaper archive, in: J. Davies, et al. (Eds.), The Semantic Web: Research and Applications, Springer Verlag, Berlin, 2003, pp. 445–458.
[15] J. Contreras, et al., A semantic portal for the international affairs sector, in: E. Motta, et al. (Eds.), Engineering Knowledge in the Age of the Semantic Web, Springer Verlag, Berlin, 2004, pp. 203–215.
[16] D.L. McGuinness, F. Harmelen, OWL Web Ontology Language overview, W3C recommendation, 2004, available at: http://www.w3.org/TR/owl-features.
[17] G. Divita, T. Tse, L. Roth, Failure analysis of MetaMap Transfer (MMTx), in: Proceedings of Medinfo, 2004, pp. 763–767.
[18] K.W. Fung, W.T. Hole, S. Srinivasan, Who is using the UMLS and how—insights from the UMLS user annual reports, in: Proceedings of AMIA, 2006, pp. 274–278.
[19] O. Bodenreider, A.T. McCray, Exploring semantic groups through visual approaches, J. Biomed. Inform. 36 (6) (2003) 414–432.
[20] E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF, W3C working draft, 2006, available at: http://www.w3.org/TR/rdf-sparql-query.
[21] Protégé an open source ontology editor, available at: http://protege.stanford.edu.
[22] V. Haarslev, R. Moller, M. Wessel, RacerPro UserGuide, 2005, available at: http://www.racer-systems.com/products/racerpro/users-guide-1-9.pdf.
[23] Jena semantic web framework for Java, available at: http://jena.sourceforge.net/.
[24] A. Bangalore, K.E. Thorn, C. Tilley, L. Peters, The UMLS knowledge source server: an object model for delivering UMLS data, in: Proceedings of AMIA, 2003, pp. 51–55.
[25] D.T. Lee, G.C. Lee, Y.W. Huang, Knowledge management for computational problem solving, J. Univ. Comput. Sci. 9 (6) (2003) 563–570.
[26] J.-F. Song, W.-M. Zhang, W.-D. Xiao, G.-H. Li, Z.-N. Xu, Ontology-based information retrieval model for the semantic Web, in: Proceedings of the IEEE International Conference on e-Technology, e-Commerce, e-Services, 2005, pp. 152–155.
[27] J. Köhler, S. Philippi, M. Specht, A. Rüegg, Ontology based text indexing and querying for the semantic web, Knowl. Based Syst. 19 (8) (2006) 744–754.
[28] H.M. Müller, E.E. Kenny, P.W. Sternberg, Textpresso: an ontology-based information retrieval and extraction system for biological literature, PLoS Biol. 2 (11) (2003) 1984–1998.
[29] A. Doms, M. Schroeder, GoPubMed: exploring PubMed with the Gene Ontology, Nucleic Acids Res. 33 (2005) 783–786.
[30] C. Friedman, L. Shagina, Y. Lussier, G. Hripcsak, Automated encoding of clinical documents based on natural language, J. Am. Med. Inform. Assoc. 11 (5) (2004) 392–402.
[31] S. Meystre, P.J. Haug, Natural language processing to extract medical problems from electronic clinical documents: performance evaluation, J. Biomed. Inform. 39 (6) (2006) 589–599.
[32] W.W. Chapman, W. Bridewell, P. Handury, G.F. Cooper, G.B. Buchanan, A simple algorithm for identifying negated findings and diseases in discharge summaries, J. Biomed. Inform. 34 (5) (2001) 301–310.
[33] D. Charnock, S. Shepperd, Learning to DISCERN online: applying an appraisal tool to health websites in a workshop setting, Health Educ. Res. 19 (4) (2004) 440–446.
[34] T.J. Errico, Syntegration: a "more complete" knowledge-based approach to the practice of medicine—North American Spine Society Presidential Address, Chicago, IL, Spine J. 5 (1) (2005) 6–12.