International Journal of Mathematics and
Computer Science, **8**(2013), no. 1, 5–23

Ontology-based Solution for Data Warehousing in Genetic Neurological Disease

Hassan Tout, Kifah Tout, Donia Awad

Department of Applied Mathematics
Faculty of Science-1
Lebanese University
Hadath, Lebanon

email: ktout@ul.edu.lb, htout@ul.edu.lb

(Received Oct. 17, 2012, Revised Feb. 2, 2013, Accepted March 11, 2013)

Abstract

In the field of genetic disorder of the nervous system, there is a huge amount of information available on the Internet. Extracting and integrating relevant information from these heterogeneous sources is a complex task usually dedicated to populate a data warehouse. Heterogeneity can be related to the structure or the semantics of sources. While solutions exist to solve the first problem, the second one remains a major problem. In this article, we propose an ontology-based solution to the problem of semantic heterogeneity of biological sources. Our ontology facilitates to build a warehouse containing data from heterogeneous sources by considering solutions in the CIG (Cooperative Information Gathering) domain. In addition to resolving the heterogeneity problem of sources, our ontology helps the built data warehouse to provide a cooperative answers to user's questions. Our solution consists of a set of three models: "Topic", "Semantics", and "Cooperative answer"; represented theoretically by logical predicates. In this paper, we present an implementation of this ontology in Protégé using OWL and SWRL languages.

Key words and phrases: Data warehouse, semantic heterogeneity, information gathering, ontology, cooperative answer.
AMS (MOS) Subject Classifications: 68T30,68T01

1 Introduction

Biological data sources are known for their heterogeneity in many aspects such as data structure and semantics. But the semantic heterogeneity is considered as the most important problem in biological database systems because it involves the content of information and its intended meaning [1].

This problem has become more and more important in data warehousing where this topic encompasses architectures, algorithms and tools for bringing together selected data from multiple databases or other information sources into a single repository called a data warehouse, suitable for direct querying and analysis. To manage this problem, the meaning of interchanged information has to be understood across the systems [2].

In this article, we propose a solution to the problem of semantic heterogeneity of data sources referring to genetic Neurological disease. Our choice of this domain is justified by the existence of a huge number of data sources referring to genetic Neurological diseases, and by the exuberance of semantic heterogeneity among their related terms. Furthermore, to our knowledge, there is not any single study that has dealt with the problem of semantic heterogeneity in this domain.

In section II, we present the main problem of building a data warehousing, i.e. "semantic heterogeneity" of information sources, and the solutions proposed to solve it. As we will see, the common point between the solutions we present is to use ontologies to deal with the problem of semantic heterogeneity.

In the second stage, we propose an ontology to deal with the problem of semantic heterogeneity when integrating data from heterogeneous data sources to build a data warehouse in genetic neurological diseases. Our ontology is made up of three models: a topic model, a semantic model and a cooperative answer model.

The semantic model deals with the heterogeneity problem, while the cooperative answer model provides a cooperative answer in response to a data warehouse user's question. The topic model contains the basic terms of the domain that are used in both semantic and cooperative answer models.

Finally, we present an implementation of our ontology with Protégé using its associated languages OWL and SWRL.

2 Semantic heterogeneity : problem and solutions

Semantic heterogeneity occurs when the same information is represented by different expressions in various sources (synonyms), or when an expression is used in various sources to represent different information (homonyms). The existence of homonyms and/or synonyms in different data sources causes a very hard problem when using these sources in a computerized integration of data from these sources to build a data warehouse.

Three main approaches exist to deal with this problem. The common point between these approaches is the use of ontologies that seems to be essential [3].

Before presenting these three approaches, let us give a definition of the term *ontology*. Here, we consider ontology as a "formal explicit specification of a shared conceptualization, where conceptualization is a set of concepts, relations, objects and constraints which defines a semantic model of a subject of interest" [4].

The three ontology-based approaches to deal with the problem of semantic heterogeneity of data sources are [5]:

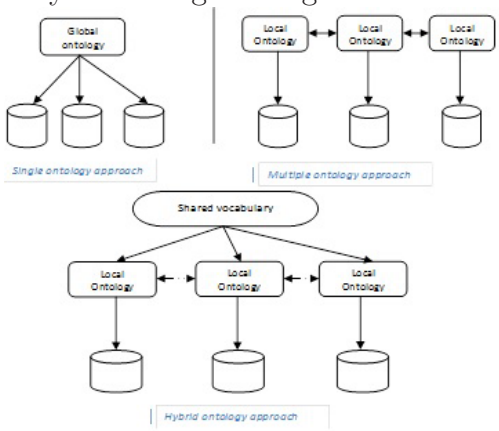
2.1 Single ontology approach

The use of a global ontology provides a shared vocabulary for the specification of semantics. All the information sources are related to this global ontology.

The global ontology can also be a combination of several specialized ontologies. This approach may be a solution to integration problems where all information sources to be integrated provides approximately the same view in a domain.

Although easy to implement, this type of ontology is not appropriate to the context of dynamic and autonomous sources. In fact, changes in one information source may affect the global ontology and its mappings to other information sources.

Figure 1: Ways for using ontologies for content explication



2.2 Multiple ontology approach

Each data source is described through its own ontology. In principle, the "source ontology" can be a combination of several other ontologies but it cannot be assumed that the different "source ontologies" share the same vocabulary.

In such an approach, the lack of a common vocabulary makes it extremely difficult to compare different source ontologies and to establish semantic relationships between them.

2.3 Hybrid approach

Similar to the multiple ontologies approach, the semantics of each source (or domain) is described by its own ontology. But in order to make the source ontologies comparable to each other, they are built upon one global shared vocabulary. The shared vocabulary contains the basic terms (the primitives) of a domain. It is to be noted that sometimes the shared vocabulary is also ontology.

3 Development of a hybrid ontology

3.1 Theoretical framework

In the domain of genetic neurological disease, we must treat a dynamic and open system where data sources may enter or leave at any time. In this context, the hybrid approach seems to be the most appropriate. Our proposition is, then, to build a hybrid ontology to define a common shared vocabulary in the domain of genetic Neurological disease.

Given the domain and the scope of the ontology, we must define different items: the terms that we are going to talk about; their relations and what is to be said about those terms.

In a first approach, we proposed to gather all the information in the domain from experts and from the related literature. Unfortunately, the number of references was too huge that it was impossible to extract consistent information. Consequently, we decided to focus on only four neurological genetic diseases that were selected as prototypes: "Epilepsy", "Autism", "Batten", and "Parkinson".

By comparing these 4 diseases, we built a list of terms that are shared by all the aforementioned diseases. It is then relevant to consider their symptoms, their diagnosis, their treatment, when they appear, and the gene disorders responsible for these diseases.

By collecting information about these terms, a list of relationships was also extracted. For example, it was found that "Batten" disease could be linked in some cases with Epilepsy. This information was then relevant to extract the disease resulting relationships. We also found more complex cases such as the fact that a person affected by Autism can have a repetitive behavior while a person who has "Batten" disease can change his behavior randomly. From this fact, we can conclude that the diseases can have opposite symptoms, etc.

Taking into account the above information, the complete ontology was built according to the shared terms and the extracted relationships.

3.2 Ontology architecture

The common vocabulary proposed is made of three parts:

1. The *topics model* that contains a set of the basic terms of the domain. This model is intended to be static and rarely modified.

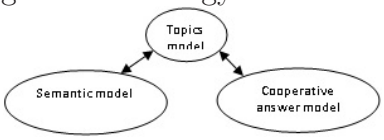
2. A set of semantic relationships that helps the system to link the basic terms from one source to equivalent terms from a different ontology linked with another data source. This set of relationships is called semantic model. This model enables us to define terms used in different sources by associating them to basic terms in the shared vocabulary.

3. The cooperative answer model that contains a set of relationships for helping the data warehouse give cooperative answers in response to the user's questions.

As "First-order (predicate), logic is the prevalent and single most important knowledge representation formalism" [14]. We propose the use of this formalism to represent the terms of our ontology, the semantic model and the cooperative answer model.

The rest of this paragraph describes the three models we identified earlier. Note that the topics model is used by the two other models as represented in the following figure.

Figure 2: Ontology Architecture



3.3 Topics models

This model contains all the basic terms of the domain. As we use the predicate logic, we define here the predicates that enable us to create the different topics. We have identified the following types of topics:

Diasease(d)	d is the name of a disease
Begin_age(d,a)	a is the age of appearance of the first symptom.
Diagnosis(d,D)	D is a diagnosis test used to detect the disease d
Dis_Gene(d,g)	Is true if the disorder of the gene g is responsible for the disease d
Symptom (d,S)	S is a symptom for the disease d
Treatment(d,T,Dmin, Dmax,u)	T is a treatment used for the disease d. Dmin and Dmax are the minimum and maximum duration of the treatment and u is the duration unit (day, month. . .)

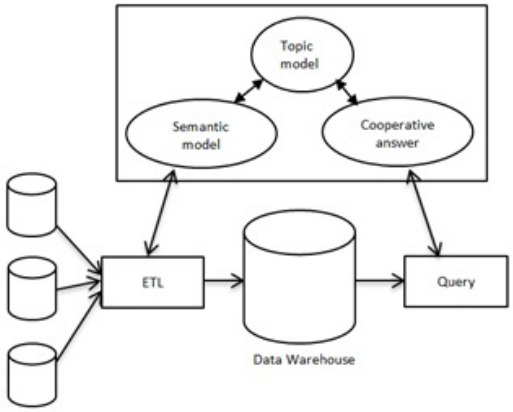
Table I. Topics

3.4 Semantic model

To face the problem of semantic heterogeneity, we propose a set of semantic relationships that use the topics identified above.

This model is used by the ETL server to define semantic links between the terms used in the data warehouse and the terms used in different data sources, and then enables the integration of data from these sources in the data warehouse.

Figure 3: Ontology use in building and querying a data warehouse



Synonym(name1, name2)	This predicate identifies two names as having the same meaning in two different ontologies or sources.	*Synonym (GRM8, MGLUR8)* In this example, the two names GRM8 and MGLUR8 designate the same gene.
TypeOf(name1, name2)	This is a hierarchical relationship that identifies name1 as a part of name2.	*TypeOf(walk , movement)* *TypeOf(talk, movement)* *TypeOf(sit , movement)* These 3 relations specify that the 3 terms walk, talk and sit are types of movements that a human can do.
Pvalue(pname, min, max, unit)	This predicate gives quantitative values to a parameter name.	*Pvalue(childage,0,13 , years)* It defines the childage as an age between 0 and 13 years

Table II. Semantic Relationships

3.5 The cooperative answer model

This model contains predicates that enable the data warehouse to give a cooperative answer to users' questions.

A cooperative answer is an answer that has 3 main properties [6]:

1. It contains only the required information.
2. It doesn't contain more information than necessary.
3. It is relevant.

The first two properties mean that the cooperative answer must contain all the information relating to the center of interest of the user and it is limited to such information. The third property involves the context of the question and means that the response must closely match with the situation in which the question is expressed.

This model is made of two parts: extensional and intentional parts.

The extensional part: this is the part where the basic objects of the domain of interest are described together with their relevant properties.

The intentional part: this is the part where objects are grouped together to form concepts, and where concepts and their properties are specified. [17]

Opposite_symptoms (S1, S2)	This predicate identifies S1 and S2 as opposite symptoms	
Disease_resulting (D1, D2)	This predicate indicates that disease D1 may lead to another disease D2.	Disease resulting (Batten, Epilepsy):

Table III. Extensional predicates

Intentional predicates are rules that are applied over ontologies to draw inferences, express constraints, specify policies, react to events/changes, discover new knowledge, and transform data, etc.

In general, rules are based on classical first order logic rules and have the form of an implication between an antecedent (body) and a consequent (head). The intended meaning can be read as "whenever the conditions specified in the antecedent hold, the conditions specified in the consequent must also hold". Both the antecedent (body) and the consequent (head) consist of zero or more atoms. Multiple atoms are connected with the conjunction operator [7].

Rules we have implemented are:

-Same_Diagnosis(D1,D2):-
-Diagnosis(D1, Diag1),

-Diagnosis(D2,Diag1)

This predicate takes as parameter 2 diseases D1 and D2 and describes them as having the same diagnosis. It enables the data warehouse to answer a question about a disease D by giving additional information on other diseases having a common diagnosis with it.

-Different_Diagnosis(D1,D2):-
-Diagnosis(D1,Diag1),
Diagnosis(D2,Diag1)

This predicate takes 2 diseases D1 and D2 as parameters and means that D1 has at minimum a diagnosis that D2 doesn't have.

-SameAllDiagnosis(D1,D2):-
Different_Diagnosis(D1,D2),
Different_Diagnosis(D2,D1)

This predicate takes as parameter 2 diseases D1 and D2 and describe them as having the same diagnosis.

-Include_Diagnosis(D1,D2):-
-Different_Diagnosis(D1,D2),
Different_Diagnosis(D2,D1)

This means that D1's set of diagnosis includes those of D2.

-Opposite_Dsymptoms (D1, D2):-
-symptoms(D1,S1),
-symptoms(D2,S2),
-opposite_symptoms(S1,S2)

This predicate takes as parameters 2 diseases D1 and D2 and describes them as having at minimum two opposite symptoms.

This model is used by the query processor to provide a cooperative answer to a user's question by adding some data or by summarizing the answer after analyzing the results and the corresponding predicates in the cooperative answer model (fig.3).

Note that, to use this model we need a query processor able to perform this analysis without affecting the query performance.

4 Implementation

In this section, we outline and justify the major design decision that has been made in the development of our ontology, mainly the ontology tools and the representation language we used.

4.1 Choice of tools

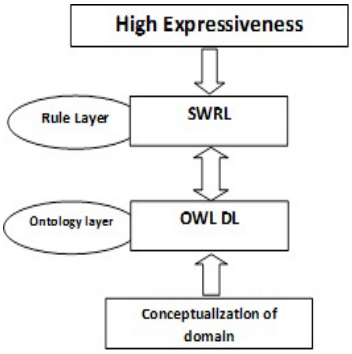


Figure 4: Representation of Built-in languages in Protégé

In the last years, the number of tools for building ontologies developed both by American and European communities has grown dramatically. When a new ontology is going to be built, several basic questions related to the tools to be used arise: What tool(s) give support to the ontology development process? How are the ontologies stored (in databases or ASCII files)? Does the tool have an inference engine? Do tools have translators to different ontology languages? What is the quality of the translations? How can applications interoperate with ontology servers? Etc. [12]

Considering the comparison done in Onto Web Consortium [12], we decided to use Protégé. Protégé is a tool developed at Stanford University for knowledge acquisition. It enables the user to build an ontology for the semantic web particularly in the W3C's by using it's built –in languages such as: Web Ontology Language (OWL) and its extended language for rules (SWRL), and for queries (SQWRL).

Web Ontology Language OWL [9, 10] was adopted by the W3C as a standard for representing ontologies on the web. OWL is a very expressive ontology representation language which can be described as a fragment of first-order logic. It comes in three different variants, namely OWL-Lite,

OWL-DL and OWL-Full, with differences in expressiveness and reasoning complexity.

As compatibility with standards was one of our major design goals, we decided to model our ontology in an OWL-compliant manner. In particular, we decided to choose a subset of the OWL language, called Description Logic (OWL DL). We argue that OWL DL provides a basic ontology modeling paradigm which meets most of the requirements above while being a flexible choice for future developments, as it is not only a proper fragment of OWL, but also of logic programming languages such as F-Logic [8].

A Semantic Web Rule Language SWRL. The basic idea of this language is to extend OWL DL with a form of rules while maintaining maximum backwards compatibility with OWL's existing syntax and semantics. To this end, SWRL adds a new kind of axiom to OWL DL, namely Horn clause rules, extending the OWL abstract syntax and the direct model-theoretic semantics for OWL DL [11] to provide a formal semantics and syntax for OWL ontologies including such rules.

Certain queries on OWL ontologies can be difficult to express, because of OWL and SWRL's open world assumption: Many types of questions require closure operations for correct formulation. For this reason, we finally have used sqwrl language.

Semantic Query-Enhanced Web Rule Language SQWRL is a SWRL-based language for querying OWL ontologies. It provides SQL-like operations to retrieve knowledge from OWL.

SQWRL is defined using a library of SWRL that effectively builds a query language on top of SWRL. These built-ins are defined in the SQWRL Ontology.

SQWRL queries can operate in conjunction with SWRL and can thus be used to retrieve knowledge inferred by SWRL rules [13].

It can also operate on known individuals in the currently loaded OWL ontology. It is very important to note that SQWRL does not provide any way to access the information it accumulates from within a rule; so, query results cannot be written back to the ontology. In other terms, they do not perform any ontology modifications.

4.2 Ontology Implementation

Our ontology contains 3 models. This section shows how to implement these models using Prot*égé* and its built-in languages mentioned above.

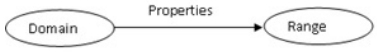
In Prot*égé*, the ontological terms are considered as classes.

These classes are supposed to be disjoint classes so that a class or an object cannot be an instance of more than one of these classes.

Besides, we defined an abstract class to be applied in case a relationship can be shared with all the classes.

Relationships are represented in Protégé as properties. Each property has a domain and a range: There are three types of properties in Protégé:

Figure 5: Properties form in Protégé



1. Object properties :This relates a class to another class.
2. Datatype properties : This relates class to an XML Schema Datatype value or an rdf literal
3. Annotation properties : This can be used to add information (a meta-data – data about data) to classes , individuals and objects/datatype properties.

The intentional predicates in the cooperative answer model (see section 3.5) are presented in Protégé as rules using SWRL:

SWRL rules are of the form of an implication between an antecedent (body) and consequent (Head)[7].

Informally, a rule may be understood as meaning that if the antecedent holds (is "true"), then the consequent must also hold. An empty antecedent is treated as trivially holding (true), and an empty consequent is treated as trivially not holding (false).

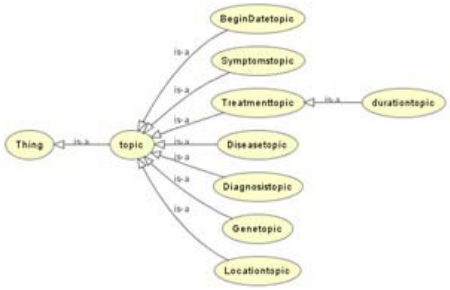
4.3 Example of implementation

Classes, like Disease, Begin age, Diagnosis. . . , are represented in Fig 5.

Relationships, as Diagnosis (d,D) in the Topic model and the Synonym in the Semantic model, are considered as object properties which relate a class to another:

1. Diagnosis (d, D): relates Disease D to its Diagnosis d.
2. Synonym (name1, name2): relates 2 classes that are the same.

Figure 6: Classes in Protégé



Complex relationships as Pvalue (pname, min, max, unit) are considered as annotation properties that imply many Data type properties:

1. HasMin
2. HasUnit

The property composants becomes:

1. Domain: topic
2. Range : topic and HasMax some anyType and HasMin some anyType and hasUnit some XMLLiteral.

For Rules , the intentional relationships have been implemented as follows:

1. Same_Diagnosis(D1,D2):-Diagnosis(D1,Diag1),Diagnosis(D2,Diag1) : can be easily represented by SWRL as :

   Diseasetopic(?x) , Diseasetopic(?y) ,

   HasDiagnosis(?x, ?a) , HasDiagnosis(?y, ?a)

   -> Same_Diagnosis(?x, ?y)

2. Different_Diagnosis(D1,D2):-Diagnosis(D1,Diag1), Diagnosis(D2,Diag1) : become wih SWRL as follows:

Diseasetopic(?y) , Diseasetopic(?x) ,
HasDiagnosis(?x, ?a) , HasDiagnosis(?y, ?b) , DifferentFrom(?a, ?b)
-> Different_Diagnosis(?x, ?y).

5 Exploitation and Discussion

To get information from the ontology, one of the solutions is to build and to analyze a request with the query language SQWRL.

As mentioned before, this language is an SWRL-Language for querying owl ontologies. It provides SQL like operations to retrieve knowledge from owl.

In this section, we present 2 examples of the queries used, each one having a different purpose:

The first query is used to test the rule that classifies the diseases having the same diagnosis.

Diseasetopic(?x) , Diseasetopic(?y) ,
HasDiagnosis(?x, ?a) , HasDiagnosis(?y, ?a)
-> Same_Diagnosis(?x, ?y)
The query can be written in 2 ways:
Same_Diagnosis(?x,?y) -> sqwrl: select(?x,?y)
Or
Diseasetopic(?x), Diseasetopic(?y),
HasDiagnosis(?x, ?diag), HasDiagnosis(?y, ?diag)
-> sqwrl: select(?x,?y)

This query can be interpreted as "If a Disease x has a diagnosis diag and a Disease y has the same diagnosis diag" (using the same variable) then the predicate is "true". It results in a list of diseases that is shown in table 1. Consequently, after performing a diagnosis, the doctor will be able to limit the number of diseases that the patient may have.

The problem with this query is that it can also consider ?x to be equal to ?y. Consequently, it presents a given disease as the very same disease with the same diagnosis:

Epilepsy	Epilepsy
Epilepsy	Batten
Batten	Epilepsy
Batten	Batten
Epilepsy	Epilepsy
Batten	Epilepsy
Epilepsy	Batten
Batten	Batten
Autism	Autism

Table IV. Result of the first query.

The second query is more complex. To support such a complex query, SQWRL has a collection of operators that provides advanced grouping and aggregation functionalities, and limited forms of negation as failure and disjunction.

This query provides more information from ontology, that we cannot have by using OWL and SWRL.

We assume now that we need to know the disease that has the shortest duration of treatment for each disease. A query to solve this situation will be written:

Disease(?x) ∧ HAS_Treatment(?x, ?tr) ∧

duree(?tr, ?d) sqwrl:makeSet(?b, ?d) ∧ sqwrl:groupBy(?b, ?x) sqwrl:least(?ld, ?b) ∧ swrlb:equal(?ld, ?d) ? sqwrl:select(?x, ?tr, ?d)

This query can be interpreted as follows: "For each disease (?x); compare the duration (?d) of treatments (?tr) is the min duration or not". The makeSet is provided to construct a set. The first argument specifies the set to be constructed and the second specifies the element to be added to the set.

The result of this query determines the minimum number of days of treatment for each disease. This can be very useful because each disease has different kinds of treatments depending on the gravity of the patient's disease. The fastest treatment can be very important to save the life of the patient.

Autism	Anticonvulsant_medication	3
Batten	Gene_Therapy	9
Epilepsy	Gene_Therapy	9

Table V. Result of the second query

These queries were tested on a small number of data. Practically, more data will be necessary to test the efficiency and the performance of these queries, and to define more complex queries. These queries will be required to answer the different intentions of the user.

Also, the ontology built is too general. Overall, this ontology has the objective of providing relevant information about neurological genetic disease to the users in response to his questions. For this objective, we have built an ontology that describes these diseases clinically and responds only to the needs of the doctors. The experts in this domain are more interested in the genetic side. In this context, we proposed to get deeper into the genetic domain and to extract information about genes, their functions, and their mutations... That will be important to get more information about the real cause of the appearance of each disease.

Finally, the built ontology must be always updated with any relevant changes as this ontology aims to gather information from different open data sources which are a dynamic, or in which the information changes every second.

For these problems, several propositions are suggested in the next version of this ontology.

For the first problem, as mentioned before, testing the queries on bigger data sources and defining new queries will be adequate.

For the second problem, new classes should be integrated in the ontology to face the needs of the researchers in this domain. Or as mentioned above in the genetic domain, there are many works that aim to solve the same problem. It is thus important to link this ontology with other ontologies of the domain like Gene ontology (GO)[15], which gathers information about the role of gene products and its relations with an organism; or OBO [16] which is an open source and aims to gather all the ontologies that describe diseases, whatever their types are.

For the last problem, we propose to integrate a rule engine in order to modify the ontology in relation with how the change of the information influences the ontology structure or its classes.

6 Conclusion

To face the problem of semantic heterogeneity of information sources when building a data warehouse in neurological genetic disease, we propose the construction of hybrid ontologies. This solution is the most appropriate in the context of open system where sources may enter or leave the system at any time and where sources are autonomous and dynamic.

We propose then to create a shared vocabulary ontology that is composed of three models:

1. Topics model that contains the basic terms of the domain. This model is used by the other two models.

2. Semantic model that contains a set of relationships that semantically link the terms in the shared vocabulary to terms used in sources or source ontologies.

3. Cooperative answer model that is composed of relationships helping the data warehouse to give an appropriate answer to the users' question.

The three models are represented using a predicate logic language, and implemented using OWL DL and SWRL languages associated with Protégé.

Ontology is a solution that seems to be the most consistent in solving the semantic heterogeneity problems, in both domains of data warehouse and knowledge base in all fields and especially in the biomedical one. We presented a simple example but we hope that this implementation will be continued in the future in order to satisfy all the needs of the experts and researchers in the biomedical domain.

References

[1] S. L. Cao, Semantic search among heterogeneous biological databases based on gene ontology, *Acta Biochim Biophys Sin (Shanghai)* 2004, 36:365-370.

[2] Jennifer Widom, Research problems in Data Warehousing, proc. 4th Int. Conf. on Information and knowledge Management (CIKM), 1995.

[3] E. Mena et al., Observer: An Approach for Query processing in Global Information Systems based on Interoperation across Pre-existing Ontologies, *Distributed and Parallel Databases Journal*, 1999.

[4] P. D. Karp et al., XOL: An XML-Based Ontology Exchange Language, February 2000.
http://www.ai.sri.com/cgibin/pubs/list_document_object.pl
?doc_uri=/pubs/technotes/aic-tn-1999:559.

[5] H. Wache, Ontology-based integration of information – a survey of existing approaches, Ontology and Information Sharing, **47**, 2001, 108–117, Seattle, WA.

[6] H. P. Grice, Logic and conversation. In P. Cole J. L. Morgan, Syntax and semantics, **3**, 1975, 41-58, New York.

[7] Millan V. Milanovie, Modeling Rules on the Semantic Web, University of Belgrade, 2007.

[8] The SWRC Ontology-Semantic Web for Research Communities York Sure, University of Karlsruhe, Institute AIFB, D-76128 Karlsruhe, Germany.

[9] Web ontology language (OWL). www.w3.org/2004/OWL/ (2004)

[10] G. Antoniou, F. van Harmelen, Web Ontology Language: OWL. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer (2004)

[11] Peter F. Patel-Schneider, Patrick Hayes, Ian Horrocks, OWL web ontology language semantics and abstract syntax. W3C Candidate Recommendation, August 18, 2003.

[12] OntoWeb, Ontology-based information exchange for knowledge management and electronic commerce IST-2000-29243, May 31, 2002.

[13] Michael Gruninger, A first-order ontology for Semantic Web Services, May 3, 2005.

[14] Stephan Grimm, Pascal Hitzler, Andreas Abecker, Knowledge Representation and Ontologies. In Rudi Studer, Stephan Grimm, Andreas Abecker (eds.),Semantic Web Services: Concepts, Technology and Applications, Springer, Berlin, 2007, 51-106.

[15] R. Stevens, C. A. Goble, S. Bechhofer, Ontology-based Knowledge Representation for Bioinformatics, *Brief. Bioinform.*, **1**, 2000, 398–414.

[16] The Open Biological and Biomedical Ontologies (OBO). Available at http://www.obofoundry.org/

[17] Maurizio Lenzerini et al., Ontology Representation and Reasoning *WP8 State of the Art Ontology Management and Engineering*. State of the Art Report, 1-18.