Semantic Data Integration: Overall Architecture

Roberto Paiano, Salento University, Engineering Innovation Department, Lecce, Italy, roberto.paiano@unile.it
Anna Lisa Guido, Salento University, Engineering Innovation Department, Lecce, Italy, annalisa.guido@unile.it

Abstract

The information has always been a valuable patrimony for the information systems that every company tries to capitalize as much as possible. With the web, the amount of information is increased and several problems arise for instance for the safety of the exchanged data but also to the semantic heterogeneity: the same information is very often represented in different ways in different information systems. In this paper we present an architecture of interchange of data both within the same information system and among different information systems founded on the ontologies in order to overcome the problem list of the heterogeneity. Ontologies, today, they seem to be the best tool useful to resolve the problem of heterogeneity, but that has not now been exploited fully.

Keywords: Semantic Data Integration, Information Systems, Ontology, Data interchange

1. Introduction & background

An information system is defined as "the set of the procedures and the infrastructures that support and describe the flowing of the information inside an organizational structure" [1]. In an information system, the "information" represent a cornerstone element and, as soon as they increase in number and complexity, increases the complexity in the retrieval and in the management of the same. Even more complex becomes the problem when the necessary information to the information system are in more source inside and/or outside the company. Very often, in fact, the same information can be in several databases within the information system, but they are represented in different ways; the same problem list arises when the same information belongs to data sources system coming from different information systems. You for instance think to a banking information system that needs present information in the tributary registry to the goal to get all the useful information to define a customer. It is important, therefore, that the company opens itself to sharing with other companies the information of interest. It born therefore the problem of the integration of this information with the goal to simplify and speed up the search task. To realize a matching of the information doesn't mean, however, to realize a comparison among string type, but it is necessary "to interpret" the meaning of every data type with the goal to semantically individualize some mappings among equal data but represented in different ways. With the advent of the semantic web [2] and of the technologies that allows the realization of this ambitious project, it is possible to delegate part of this job of "interpretation" of the meaning from the human to the computer increasing in this way the efficiency of the system and decreasing the necessary time to realize this matching as well as the probability of error. The Web provides the possibility to share a myriad of different data source; the diffusion of the XML standard as syntax of the shared data, have facilitated besides the process of sharing of the information. The semantic data integration represents a process able to automatize the communication among different systems providing a realistic design of the meaning of the data of these systems. This process, mainly founds him on a search of the semantic relationships (existing or derived) inside the metadata of the systems. Despite the semantic web (and the relative technologies) has given a strong impulse to the semantic integration of data coming from heterogeneous source data, the semantic integration is an extremely difficult problem. We consider, for instance, the difficulties that arise during a process of scheme-matching, that is of the search of semantic correspondences (you call matches) among schemes of database. Mainly, the matching among two schemes of database sets the problem to decide when two elements of different schemes belong or not to the same concept in the real world and in this the technologies of the semantic web help quite a lot. Several has been the attempts of use of the ontologies to join data coming from heterogeneous sources. Among these we quote [3] in which are defined some semantic affinities among the ontologies coming from several relational models using some established semantic weights and a special engine that, through mechanisms of inference, it extracts one (or more) ontologies of domain shared among the different data sources. Another interesting approach is proposed in [4] where authors propose an algorithm of extraction of ontologies starting form a relational database.

The extraction process is followed by a matching among the ontologies so gotten and the ontologies of domain made up in a separate way. The problem of the integration of data coming from heterogeneous sources is a hard problem, especially in the public administration [5] where information are distributed on several data bases that can be physically within a same information system or on different information systems.

1.1 The multi agent system

Through the multi agent systems (MAS) it is possible to model complex systems such as systems where it is important the exchange of data coming from the same or from different information systems. These agents can indirectly interact among them (acting on

the environment of domain) or directly (through the communication and negotiation). The agents can decide to cooperate for a common benefit or to face his/her own problems. Then, an agent is

- Autonomous: it operates without the direct intervention of the human or other systems;
- Social: it cooperates with the human or other agents to perform it's own tasks;
- Reagent: it interprets his/her own domain, and it answers in a determined range of time to the changes needed in the same domain.
- Proactive: it doesn't simply act in answer to its environment, but it is also able to exhibit a behavior goal-directed autonomously taking the initiative.

The architectures of the agents define the mechanisms thanks to which an agent acts in effective ways in a real environment, dynamic and open to external influences.

One of the key aspects of the multi agents systems it is the communication. Particularly, the agents interact among them using some special languages of communication, called Agent Communication Language (ACL), that provide a separation between the action of the communication and the language of the content. Even more interesting and large are the possible applications of the multi agent system in the Semantic Web.

A lot of different organizations are working in the realization of multi agent systems in which the technologies of the semantic Web are used to support the agents both in the search, in the filtration and in the manipulation of the information that in the composition of the business process.

In this paper, after having defined the problem list of the semantic integration of the data, we present a useful architecture for the semantic integration of the data that exploits the ontologies as tool to enable the integration. We present also a first step toward the integration of heterogeneous data sources, within the same information system.

1.2 Semantic data integration

From the preceding section, it is clear that the semantic web, opportunely integrated in a multi agent logic is a useful tool for the integration of data coming from heterogeneous data source (both within the same information system and in different information systems). Before exposing the architecture, it is useful to put in evidence some aspects. First of all, it is important to individualize the assistant value that the use of the ontology provide in comparison to the use of the database. The database, is them relational or object oriented, have the main task to memorize and to organize the information, and they make available of whom uses them all the information of which it requires.

The ontologies, are more oriented toward a more expressive description of the data or the information and the information it is also defined complete when it can be determined beginning from another information.

The ontologies result very more flexible than the schemes of the database: in the ontology it is possible to omit some information that will be deduced in a second moment by the context in which they operate. This requires, however, the use of mechanisms of inference also very complex. Another important observation comes from the fact that the information in the database they derive from its schema and from the integrity constraint. If the scheme doesn't foresee the storage of some information they cannot be memorized; likewise if information violate the integrity constraints they must not be stored. The ontologies, instead provide a further level of flexibility: ontologies allows the storage of information in arbitrary ways unless something (that is any constraints explicitly taken) doesn't prevent such an association of it. From this it is possible to say that in a very dynamic environment which is that allows the sharing of information among different information systems it is worthwhile to exploit to the best the semantic power of the ontologies leaving the data management to the by now mature by DBMS (database management systems). This brings as immediate consequence to the fact that the integration has to necessarily happen to level of metadata. Useful would be, in fact, an integration to the data level (and not at the metadata level) but the technologies of the semantic web are not still enough mature to allow the management of a knowledge base useful to memorize the information coming from heterogeneous data: it would be gotten in fact a knowledge base that, also containing all the necessary information, it would be of very full-bodied dimensions and therefore hardly to manage. To this point it is important to understand if the integration has to be completely automated or if the human intervention is necessary to facilitate and to improve the mechanisms of integration. Surely, a completely automated mechanism could bring above all to wrong attributions of meaning when the application domain is of the niche, or rather it contains terms commonly not used.

It is important, therefore, that this process of mapping is made up with the aid of the end user but without taking the risk to continually ask its intervention. To face this problem, particularly useful seem to be Wordnet (http://wordnet.princeton.edu/). Wordnet is a semantic lexicon in English language that provide a lexical database able to semantically connect all the words (names, adjectives, verbs etc) providing a classification in base to all the possible meanings correspondents. Wordnet is particularly profit to individualize synonymous of the words in the database: the synonyms are particularly useful with the goal to individualize a mapping among metadata in rising different data but with analogous meaning. Parallelly to the aid of Wordnet it results particularly profit to support the system on one (or more)

ontologies of domain that opportunely describing the context, of aid in the difficulty assignment to individualize analogous concepts but otherwise represented in different databases. Using from a side Wordnet and from the other one the domain ontology, it is possible to reduce as minimum as possible the number of interrogations done to the consumer, and it is possible to make only reference to the end user to solve possible problems for instance tied to the choice among a set of terms suggested following the searches done on Wordnet and/or on the ontology of domain. Verified the importance of the use of the ontologies with the goal to facilitate the semantic integration of the data, it is important to individualize an ontological language useful to represent the ontology. It is possible to chooses, therefore, the ontological language OWL by now standard W3C and that currently is passing to the version 2.0 [6] that, still in draft, it represents an extension of the preceding version. Owl 2.0 will introduce new characteristics what extra syntactic sugar, additional property and qualified cardinality constructor, extended datatype support, simple metamodelling and extends annotation. With owl 2 will be possible to realize in simpler way the reasonings of inference. The absence in international scientific circle of solid and efficient tools of inference has represented, thin to today a bottleneck in the development of the ontologies.

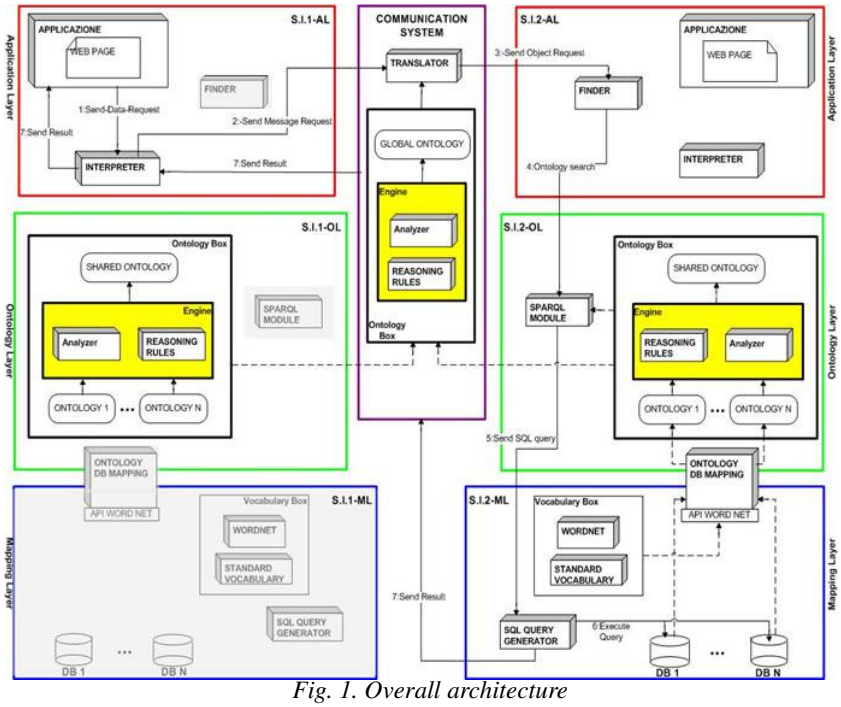

Fig. 1. Overall architecture

2. Proposed architecture

Starting from the considerations done in precedence, in this paper we present an architecture for the data sharing among heterogeneous information systems. The heart of the architecture consists in the presence of the ontologies that simplify the semantic matching of the information. The system is seen as a multi agent system in which every information system has the role of agent with an own operational and decisional autonomy, but that it cooperates with the other agents with the goal to get the information of interest. The architecture is made up of two (or more) information systems and of a communication system with the role to allow the communication among them. It is important to note that all the module inside the architecture are present within a single information system: different colours have exclusively been underlined here for improving the legibility of the figure. Every information system is structured on 3 layers:

- **Mapping Layer** (in low) with the task to extract, through a standard vocabulary and/or through Wordnet, the metadata from the several database inside the information system.
- **Ontology Layer** (in the central part): it deals particularly with the management of the ontologies, applying a special methodology that is not the goal of this paper, the layer make the merge of the ontologies gotten beginning from the data sources.
- **Application Layer** (in the high): it constitutes the interface with the end user to it has the double assignment to send the useful information to affect the search of a data of interest and to receive (and eventually to elaborate) the result of the search.

The communication layer has the goal to manage the communication among the different information systems. As it will be clear subsequently it is made up from two macro modules one (translator) with the goal to interpret the message coming from the information system converting the message in the format also comprehensible easily from the other information systems. The second block, analogous to that present in the ontology layer within the information system, has the goal to realize a global ontology beginning from the shared ontology that are in every information system. In the global ontologies, will be present also useful information to route a specific request in the specific information system that will answer to the request. In practice the ontologies of the information system that belongs to the same domain, are elaborated through particular algorithms of 'merging' in order to get an only ontology called *Shared Ontology*. The Shared Ontology contains a set of common terms, properties and existing relationships and shared by all the participants to the system of communication, and it is useful note to affect searches in the different ontologies to which ago reference, using semantic criterions. Equally, on the system of communication there is a *Global Ontology* constituted by the merging of the Shared Ontology coming from the information systems of the domain of affiliation. The Global Ontology is used then as general indexes, to which will make reference the system of communication, for the forwarding of the requests.

We now describe each module that constitute the architecture here introduced: subsequently, we will define as the various blocks are composed.

- **Application:** it represents the application (typically a web-application) used by the user to the goal to search for a particular data. The search criterions are inserted in a request sent to the Interpreter module. The dispatch of this request is labeled in figure as *1:Send-date-Request.*
- **Interpreter**: this module has the goal to serve as medium between the application and the system of communication. Particularly it will work in two directions:
  - From the application to the system of communication: it has the assignment to translate the applications of the application in a comprehensible message from the agents (for instance a message ACL), and to forward him/it to the translator on the communication system (*2:Send-Mesage-Request*).
  - From the system of communication to the application: the Interpreter also has the goal to translate the messages from the system of communication, in a language that can be understood from the application, and to forward them to this last (*7: Send Date Request*).
- **Translator**: it receives in input the request from the Interpreter, it consults the Global Ontology, and it sends the applications translated to the module Finder of all the information systems *(3:Send Object Request)*.
- **Finder**: it maintains a queue of the messages coming from the Translator, and it dispatch them the SPARQL Module *(4: Ontology search)*, activating the search on the ontologies.
- **SPARQL Module**: it made the search on the ontologies through query SPARQL, and it passes the search criterions to the SQL query generator *(5: Send SQL query)*.
- **SQL Query Generator**: it gets in input the search criterions, and it produces a query that execute (*6: Execute query*) in order to extract the possible data from the database. The answer is sent to the applicant (*7: Send Result)*.
- **Analyzer**: it exploits the reasoning rules to fill the Shared Ontology or the Global Ontology.
- **Vocabulary Box:** it contains the dictionaries that will be used for the mapping database-ontology, and for the search in the database of possible synonyms of the search key, in the case in which this last was not found. The Vocabulary Box contains two blocks, correspondents to two different dictionaries: WordNet and Standard Vocabulary that is a domain ontology that can be added in optional ways from the administrator of the system.
- **Ontology Box**: it contains all the ontologies of the information system and the Shared Ontology. The shared ontology is an ontology that contain relationships of equivalence among the different ontologies of the same application domain, produced thanks to the aid of a reasoning engine.
- **Reasoning Rules**: has the task to make inference on the information in the ontology. It starts from the rules defined by the system administrator.
- **Ontology DB Mapping:** it makes a seed-automatic mapping of the database in the ontology that describe the structure of it.

2.1 Details of the architecture

Each module in the architecture is made up of several modules. We describe in details each of them. The order of the description follows the number of the arrows.

Application

The module *Application*, in the application layer has the goal to start up the request of data. In order to do so, it is possible to use a generic web page that send, using the http protocol, the request to the module *Interpreter*.

Interpreter

The module Interpreter, in the application layer, is made up of the following components:

- **HTTP Receiver:** it receives the request of the application;

- **Application Agent for Request**: it has the goal to manage messages, and it is made up of two modules:
  - **Message Translator**: it read the request coming from the *http Receiver*, and it made a message to send to the "Ontology of Request";
  - **ACL Sender**: it sends the message in the ACL format.

Translator
The translator module (in the communication system) is made up of:
- **Application Agent Searcher**: it obtains the request coming from the interpreter, and it has two components:
  - **Message Request Receiver:** it receives the message from the interpreter
  - **Message Request Sender:** if the search in the Global Ontology provides positive results, the *Message Sender Request* send the message to the information system that contain the data, otherwise the message will be sent in broadcast.
- **Ontology Query Translator**: it translates the message in a SPARQL search query, and it send the message to the *Ontology Searcher*;
- **Ontology Searcher**: it executes the query on the Global Ontology made up by the Ontology Query Translator, and it sent the data obtained from this ontology to the Message Request Sender.

Finder
The Finder (in the application layer) is made up of :
- **Application Agent Receive Request**: it obtains the request coming from the communication system.
- **Ontology Search Activator**: it stores the queue of the messages coming from the *Application Agent Receive Request* (and it establishes the priority) and it foreword these requests to the SPARQL Module.

SPARQL Module
The SPARQL Module (in the ontology layer) is made up of two modules that is:
- **SPARQL Translator**: it receives as input the criterion for the query generation, and it generate the *SPARQL query* that will be sent to the *SPARQL Query Manager*;
- **SPARQL Query Manager**: it obtains a SPARQL query form the SPARQL translator, and it execute these queries on the ontology. In a second phase, it will send the result of the query to the *SQL Query Generator*.

SQL Query Generator
The SQL Query Generator in the mapping layer is made up of the following modules:
- **SQL Translator**: it receives the possible location of the data (provided by the SPARQL module), and it generates an SQL query.
- **SQL Query manager**: it receives the query form the *SQL Translator*, and execute it in the relative database and send the results to the *Application Agent Result Sender*;
- **Application Agent Result Sender**: it translates the message with the result of the SQL query in a language simple to understand from the communication system, and it send it. This module is made up of two modules: the *Message Translator* and the *Result Sender***.**

Vocabulary Box
The Vocabulary Box (in the mapping layer) is a useful module for the mapping between database and ontology: it is made up of two main modules:
- **WordNet**
- **Standard Ontology**: that is a standard ontology that define the application domain.

Engine
The Engine module used to obtain the Shared (or Global) Ontology is made up of:
- **Analyzer**: this module has as input each ontology, and it generate the shared ontology trough a well defined methodology that is not the goal of this paper.
- **Reasoning Rule**: In this module it is possible to store and retrieve the inference rules useful to make the ontology.

3. Mapping layer

We focus in this section to the mapping layer. It is possible to obtain an ontology starting from a database and following a default series of criterions, as for instance those pointed in [7]. The mapping must consider not only the structure of the database but also the format of the instances, despite the real data must be in the same database. The choice to maintain the data in the database and not in the ontology, resides in the verification that the technologies to search data inside ontology are not enough of full age. You believe that currently, the insertion of a great amount of data inside the ontology would bring to a degrade of performances, in particular ways in phase of search. What mostly interests us, is the possibility to create an ontology that besides the aforesaid criterions also follows some criterions of classification of the names in base to a default dictionary, such as WordNet. Since it is impossible to update WordNet, if not from the same developers of the application, we think to add to the architecture an optional ontological dictionary. The aforesaid dictionary, won't be anything else other than an ontology of domain, that can be inserted previously for instance by a person responsible of the system, and update as soon as it proceeds with the mapping. The use of this additional dictionary, implicates that the search of the words and their possible structure, is effected on it, when the words themselves have not been found in WordNet. In the information system could be present more databases. With the goal to

understand this last affirmation, we see a clarifying example related to the phase of mapping. We consider, for the sake of simplicity, a database made up by an only table (the table Consumer, with attributes Name, Last name, Address). Established what element of the database we want to translate in the ontology, the name of this element will be searched in WordNet.

To this point two eventuality can be verified if the name of the element is or isn't in the Word Net. In the second case, we make a search in the domain ontology. If the name is in the domain ontology it will be inserted in the ontology otherwise the system ask the user to specify a correspondent in WordNet or in the ontology for the name considered. If the correspondent one in WordNet (or in domain ontology) it will be selected by the user and it will be put in the ontology. The original name will be put in the Object Property. If the consumer doesn't specify a correspondent in WordNet will be inserted in the ontology an object with the original name of the element of the database. In the first case, the element in the ontology will have the same name of the element in the database.

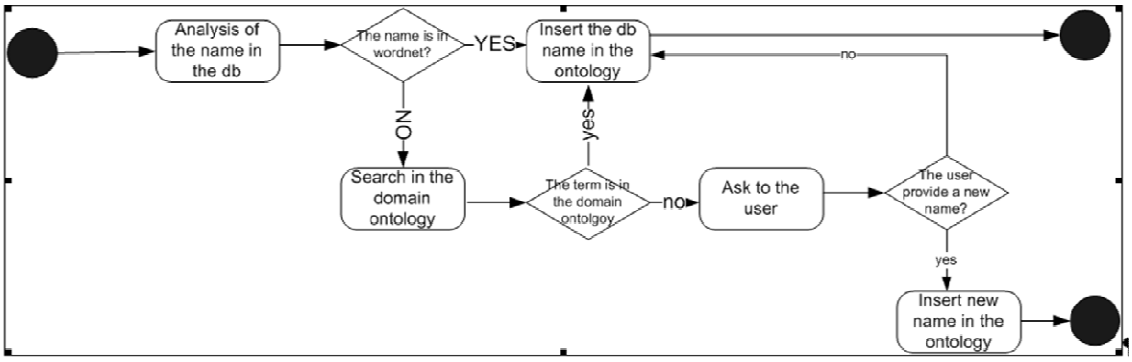

Fig. 2. Flow diagram for the mapping of terms

4. Use case

We explain with an example how the architecture works. If we consider that the information system has more than one database, we must generate a specific ontology for each of them, a Shared Ontology for each information system and a Global Ontology for the overall system. We consider to have two information system I.S. 1 and I.S. 2 that communicate between them through the Communication System (C.S.) and we think that a user in the I.S. 1 made a request through a web application. This request is the CAS number of the lys an amino acid of the vegetable and animal protein (in the hemoglobin). We suppose that the I.S. 2 has a complex data storage system made up of several databases. In this condition, although data is in a database, it is hard to search it in a few times. For this reason in the architecture, there is a mapping that allows to organize in a semantic way the information in the ontology in order to make fast the search. In this case study, the I.S. 2 has two databases: the first has all the protein classes and its amino acid (hemoglobin and lys) while the other database has only the amino acid. In the mapping the types of information in the database will be mapped by the Ontology DB Mapping module in two ontologies that has the goal to classify the information:

- ONTOLOGY 1 : PROTEIN -> AMINO ACID
- ONTOLOGY 2 : AMINOACID

The Analyzer module using the rules of the Reasoning Engine, will obtain the Shared ontology will state that

ONT-1::PROTEIN::AMINO ACID = ONT-2::AMINO ACID

After this mapping phase, a mapping in the Communication System will allow to define, thanks to the Analyzer, the Global ontology where it is possible to find information about the location of the data. We follow the flow of the figure 1 in order to understand how the search will be performed.

1:*Send Data Request* the user makes the request using the web page. The request will be sent to the Interpreter module.
2:*Send Message Request* The module Interpreter will translate the request in a message that each I.S. can understand, and it sent its message to the Translator module in the communication system.
3:*Send Object Request* The translator module will ask to the Global ontology for the PROTEIN in the Global Ontology, and it send to the Finder the message with the position of the data. if the data is not in the Global Ontology, the Translator start the search in the ontologies of the information system of the architecture.
4:*Ontology Search* The Finder module takes the message with the search criteria and the location of the data obtained by the Global Ontology, it starts the search in the ontology.
5:*Send SQL Query* The SPARQL module use the criterions from the Finder in order to generate the SPARQL query. This query will be used to search

in the ontology. In this case study, the query finds the terms hemoglobin .
6: *Execute Query* The query sql will search for the CAS number of the lys in the database of the I. S. 2.
7: *Send Result* The data will be sent to the communication system that send the data to the requested application thanks to the Interpreter module.

5. Conclusions and future works

In this paper, considered the importance that information assume within the information systems, we show an architecture based on the use of ontologies with the goal to facilitate the semantic integration of metadata and to realize a common database for more information systems that want to cooperate for exchanging information.

This introduced the first step toward the realization of this architecture: the next step to be done, and on which the research group is already working, it consists of individualizing a methodology that allows to realize the semantic matching among the extracted ontologies by the different database is them belonging to the same information system, is them belonging to different information systems.

Naturally most attention will be to the communication system that has the goal to guarantee the interchange of data among different information systems.

The design and implementation of the whole architecture will be followed by a test of the realized tools.

6. References

[1] Pighin, M., and Marzona, A.: "Sistemi Informativi Aziendali", Pearson Education, ISBN/ISSN 9788871922447 (2005).
[2] Tim Berners-Lee, J. H. "The Semantic Web" *Scientific American,*2001.
[3] Trinh, Q., Barker,K., AlhajjR.: "Semantic Interoperability Between Relational Database Systems". *Proceeding in the 11th International database engineering and application symposium* IEEE pp. 208-215.
[4] Lee,K.J. Keun Whangbo,T. "Semantic mapping between RDBMS and Domain Ontology" IEEE 2007
[5] Grosso,R.: "History of CSI Piemonte in metadata cataloguing, knowledge inference and ontologies". *In proceeding 2nd International Conference on Methodologies, Technologies and Tools enabling e-Government* 25th and 26th of September 2008.
[6] W3C "OWL2 Web Ontology Language: Primer. Working draft" 11 April 2008-10-14
[7] Wang, M. L.-Y. "Learning ontology from relational database". *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 2005.