

SEMANTIC INTEGRATION OF INFORMATION SYSTEMS

Anna Lisa Guido, Roberto Paiano

Department of Engineering Innovation, University of Salento, Via per Monteroni,
73100 Lecce, ITALY

annalisa.guido@unisalento.it, roberto.paiano@unisalento.it,

ABSTRACT

In the previous years Information System manage only information inside the company, today a company may search and manage information of the other companies. In this scenario the problem of communication between Information Systems is of the highest importance. Up to the present moment, several types of integration have been used but all were founded on the agreement (about data to share and the exchange format) between the interested Information Systems. Today, thanks to the new technologies, it is possible that an Information System uses data of another Information System without a previous agreement. The problem is that, often, the Information System may refer to the same data but with different names. In this paper we present a methodology that, using ontology, and thus the intrinsic semantic of each data of the Information System, allow to create a global ontology useful to enable a semantic communication between Information Systems.

KEYWORDS

Semantic Integration, Ontology merging, Methodology, Information Systems

1. INTRODUCTION

The competitive market where companies works bring to the necessity, for each company, to share information coming from its own Information System and to use information that another Information System provides. In this scenario, the integration of existing Information Systems is becoming more and more essential and the companies may use not only the heterogeneous data source coming from long-term investments in the IT infrastructure but also information coming from other Information Systems. One of the greatest problems to face, when we want to allow the communication among different Information Systems, it is the fact that they found themselves on the use of several data sources structured in different ways that, often, represent equivalent information. The integration problem can be faced on the 5 levels of a typical Information System [1]: User Interface, Application, Middleware, Data Management and Data. On every level it is possible to use a different approach to the integration:

- **Integration by Applications:** it uses integration applications that can access to several data sources, and they return integrated results to the consumer.
- **Integration by Middleware:** they provide some reusable functionalities that generally solve integration aspects, as for instance the SQL-middleware.
- **Uniform Data Access:** a logical integration to the data level is performed.
- **Common Data Storage:** integration of the data made up moving the data to a new source of memorization.

There are several systems that allow to manage the approaches here described:

- **Data warehouses.** They realize an approach of common data storage for the integration.
- **Systems of federate databases (FDMS).** They provide a solution for the uniform access to the data logically integrating the data from different underlying local DBMS.
- **Integration through web-services.** They realize the integration through software components (web services) that they support the interaction machine-to-machine through a net using XML-based messages.

The approach here listed impose that the data exchanged during a possible communication are agreed in advance. Therefore, it is impossible to make a search on data that are not in the "list" of data agreed. This sets a strong limit to the integration among heterogeneous Information Systems. To give a solution to this problem, in the international scientific circle it has been used the semantic data integration. The first problem to solve, within the semantic integration, it is the heterogeneity of the information both inside the same Information System and among different Information Systems. The problem is very complex, and it cannot put aside from an ad-hoc methodological approach. Methodology has to guarantee a transparent communication to the consumer, and without a previous accord about the treatment of the data among the system's participants to the interchange.

One of the key points to realize the integration among Information Systems, it is the grouping of different sets of data schema so that to be able to define a *global schema* that represents and contains all the information proper of the Information System. This global schema will be the connection with the other Information Systems. This problem, within the semantic integration of the data, it is identified with the problem of the Schema Integration that is translated in the problem related to the process of schema-matching, that is a problem within the research of semantic correspondences (named matches) among database schemas. Mainly, the matching among two schemas of database sets the problem to decide when two elements belonging to different schemas belong or not, to the same concept in the real world. A good way to make a semantic integration of data is the use of ontologies that, defined as an explicit and formal description of concepts and the relationships among them, can contribute to the solution of the semantic heterogeneity. Starting from these considerations this paper provides a design methodology for the generation of a shared ontologies that, according to the approach of ontology-matching, allows to produce a shared ontology inside an Information System that represents, starting from the data schemas inside the Information System, the semantics of the information in it contained. But the definition of a methodology that allows the integration between different ontologies coming from different Information Systems is not the only goal of this paper. We present here also an architecture that allows the communication of different Information Systems: in this architecture can be used efficiently the proposed methodology. In this paper we present in the section 2 a background related to the upper level ontology, the ontology matching and the existing distributed communication systems. In the section 3 we make an overview of the proposed system architecture and we explain the details useful to understand it. In the section 4 we detail the methodology that allows to obtain the ontology. Finally, in the section 5 we conclude our work.

2. BACKGROUND

Ontology matching is only a step of a more complex architecture used to obtain semantic integration among information systems by ontologies. For this reason in this section, in addition to the existing ontology matching approaches, we analyze the background related both to the upper level ontologies and the technologies for the communication systems.

2.1 The upper level ontologies

A powerful support to the semantic integration of the data is provided by the ontologies [2] (that can be defined as an explicit and formal description of concepts and their relationships that exist in a well specified domain using a dictionary that describes these concepts). Ontologies can

contribute to the solution of the semantic heterogeneity. In comparison to other schemas of classification as the taxonomies, the thesaurus or the keywords, the ontologies allow to represent a domain model in more complete and precise way. Through the ontology, the semantics of the information can be made explicit. An interesting approach to the development of ontologies is to use the high-level ontologies (upper level ontology) as external source of common knowledge. Examples of these types of ontologies are *Cyc ontology* [3], *SUMO* (Suggested Upper Merged Ontology) and *DOLCE* (Descriptive Ontology for Linguistic and Cognitive Engineering) [4]. The key characteristic of these types of ontologies is to provide a semantic high-level source useful to derive the meaning of objects or classes belonging to the low level ontologies. They allow besides to facilitate the communication based on different ontologies by translate the relative concepts of an ontology of high-level of reference. For instance, in the field of the anatomy it is possible to use a defined ontology *FMA* (The Foundational Model of Anatomy[5]) to allow systems based on different medical ontologies to communicate in efficient way. One of the high-level ontologies that more it is affirming in international scientific panorama is WordNet (<http://wordnet.princeton.edu/>) that it provides a lexical database able to semantically connect all the words (names, adjectives, verbs etc.) providing a classification according to all the possible corresponding meanings.

2.2. Ontology Matching

In the research about data integration, the possibility to use ontologies for the domain representation seems the best way to face the problem of the semantic heterogeneity. It is possible to use as models of domain of the information a single global ontology toward which local information will be translated. In this way, it is possible to question through query, that operates on the global ontologies, all the local sources inside an Information System. As it regards the semantic integration of the Information Systems, it is difficult to translate immediately the various domain information to only one ontology, except in the case in which these domains are very similar. In the general case of heterogeneous Information Systems, we speak about *multi-ontologies*. In this scenario, the several local sources of information must be translated in the corresponding local ontologies able to semantically characterize them; at this point, it will be possible to semantically compare the ontologies and to produce a shared ontology that represents the result of the semantic integration of such sources. Such scenario is translated in an operation of search of the semantic correspondences among the various ontologies: ontology-matching [6]. In this context it passes from the semantic integration of data represented by generic data models (typically relational databases) to a common representation realized through ontologies that subsequently will come in a unique global ontology. If data are contained inside companies databases it is important to realize a specific mapping among relational databases and ontologies: this mapping allow us to create a semantic layer over the database. Even if relational databases are based on closed-world assumption and ontologies use open-world assumption, there have specific correspondence between them. Three approaches exist about the generation of semantic layer over the legacy database:

- i. Semi-automatic generation of an ontology from a relational database. For instance an attribute in a relational database schema can correspond to a specific OWL ontology property. Indeed a relational database can be formalized by the First Order Logic [7], and OWL ontology thanks to Description Logic (DL) [8], a subset of FOL. Then it's possible to build a mapping among relational databases and ontologies, and this mapping should use a two phase paradigm:
 - a. Research of a simple mapping between relational database scheme entity and ontology;
 - b. Building complex composition based on the realized mapping.

- ii. Manual notation of a dynamic web page that publishes database content. These notes contain the information about the underneath database and the way to extract from it the page elements.
- iii. Mapping of an existing database over an appropriate existing ontology implemented in RDF or OWL, using such languages as D2R MAP [10], or Extended D2R [11].

In [12] we can see an interesting suggestion involving the exposed scenario and the first approach, where the local resource are local ontologies generated from local databases and where the authors proposes particular framework for the relational database interoperability, *Fig. 1*.

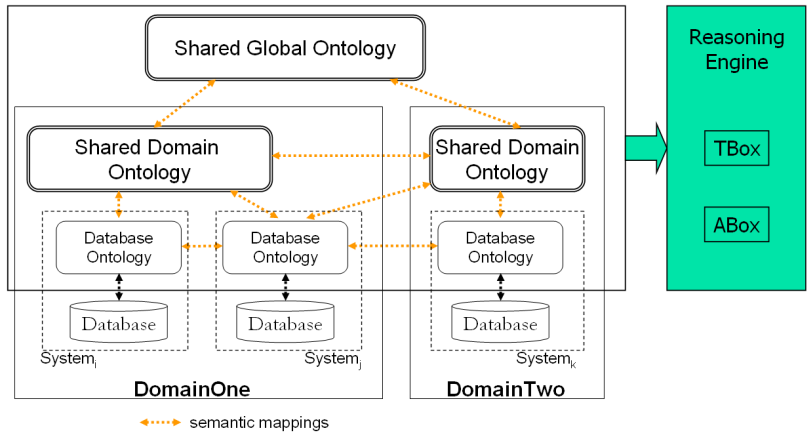


Fig. 1: RDBMS interoperability Framework

In figure 1 we can note:

- i. A common vocabularies set, containing semantic relation and constraints, that describes the RDBMS
- ii. Standardized database ontologies describing the RDBMS semantic;
- iii. An ontologies set that collect all the semantic relations among the ontologies of different databases;
- iv. A reasoning engine that deduct any semantic relations among databases ontologies. A-Box and T-Box in particular, reason both about concept and instances, and moreover they perform correct inference among input ontologies.

Semantic mapping among database ontologies, it's very important for the RDBMS interoperability, and we can define it in three different way [12]:

- i. Among database ontologies (*Fig. 2a*);
- ii. Among database ontologies and shared domain ontologies (*Fig. 2b*);
- iii. Among database ontologies, shared domain ontologies and shared global ontologies (*Fig. 2c*);

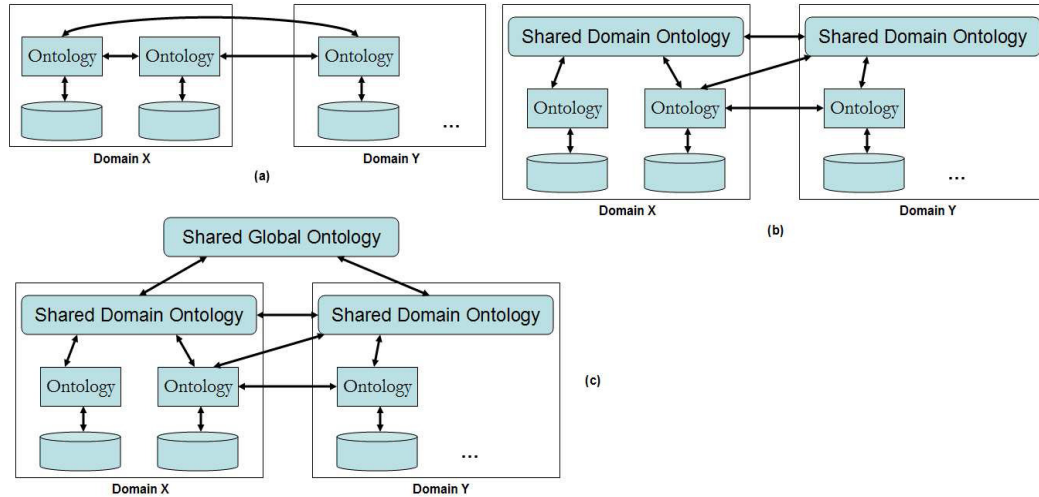


Fig. 2: Mapping ontologies layers

2.3. Communication System

The existing technological and architectural solutions in the communication field allows data sharing among Information Systems. Some interesting solutions are listed below:

- i. *SOA and SCA*: the reference model provided by OASIS [13] for SOA (Service Oriented Architecture) underlines some fundamental concepts that any implementation is based on [14]. According to the SOA reference model was created the SCA (Service Component Architecture) architectural style. SCA provides both synchronous and asynchronous communications. It has a great scalability and many languages implementations but it have not a primal ontology support.
- ii. *JADE* (Java Agent Development Framework [15]), a framework software that makes easier the development of agents applications, supplying services complying to the FIPA[16] standard and the chance to work efficiently with other platforms. JADE is a distributed Information System able to deal ontologies, that allows the communication by the exchange of ACL messages. The drawbacks of this communication system, are related to his centralized and not scalable architecture, instead his advantages are related to the high interoperability among heterogeneous systems and to the primal ontology support.
- iii. *DDS* (Data Distribution Service for Real-time System [17]): is a specification of a publish/subscribe middleware for distributed systems created in response to the need to standardize a data-centric publish-subscribe programming model for distributed systems". The drawback of DDS is the lack of a born support for the ontologies, that in our architecture plays a key role.

3. ARCHITECTURE OF INTEGRATION AMONG DIFFERENT INFORMATION SYSTEM

In figure 3 there is the general architecture [18] that allows to realize the semantic integration inside an Information System. The definition of the architecture considers two fundamental aspects: the semantic representation of the information and the interoperability. For the representation of the ontology, we chose the OWL language [19]. As it regards the

interoperability among heterogeneous Information Systems, we decided to use an agent-layer, based on the use of software agents, through which it is possible to semantically interpret the requests coming from the application layer of an Information System and to begin, therefore, a phase of search inside the Information System in relationship to the message received from the application.

The architecture is on three levels:

- **Application Layer:** it represents the highest level of the architecture. It contains a generic application that takes as input the criteria of search, and it sends a request to the communication system. It receives, finally, the results.
- **Ontology Layer:** it represents the intermediate level in which are managed, inside an Information System, the local ontologies and the shared ontology.
- **Mapping Layer:** it deals with the mapping among the data stored in the database and the local ontologies.

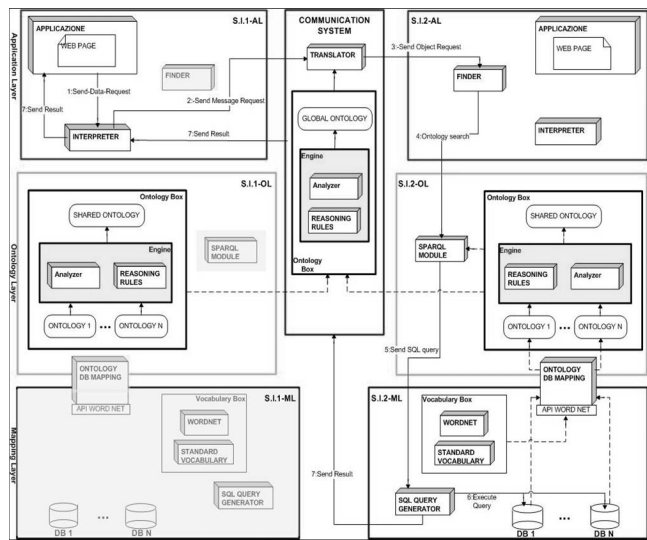


Figure 3: Overall architecture for the semantic integration between heterogeneous Information Systems.

In the next paragraphs we will describe in details the most important architecture components and modules.

3.1 Application Layer

The Application Layer contains the web-application employed by the user to search particular data. This application requires data using ACL Messages that is delivered to the interpreter module. The Interpreter link communication system with the application, and translate messages from ACL to Standard message language or vice versa.

The Finder module maintains a message queue coming from the interpreter and send them to the SPARQL Module according to FIFO logic.

3.2 Ontology Layer

The Ontology Layer is located both on the Information Systems involved in the communication process and over the communication system. Here is generated a shared ontology that contains the equivalence relations among all the ontologies of the same applicative domain. This ontology is created by the help of a particular matching algorithm[20].

Two modules are located inside this component: ontology box and SPARQL module.

Ontology Box

This module is located inside the Information System and take in input all local ontologies, to give back in output a shared ontology. Ontology Box is also located inside the communication, but here it takes in input the shared ontology sent by different Information Systems of its domain, and it returns in output a global ontology.

The Ontology Matching Algorithm generates a shared ontology constituted by the merging of local ontologies, and represents semantic equivalent classes in these ontologies, defining shared information located in different sources of an Information System. This algorithm is explained in [20].

The Shared Ontology work is to improve the information research over the ontologies, acting as a Yellow Pages service, that stores reference about all mapped ontology classes. This is the reason of sending to the communication system where it helps to create the global ontology.

The Global Ontology created over the communication system, also store the reference to all domains shared ontology.

SPARQL Module

The SPARQL Module automatically generate SPARQL query to interrogate the ontologies located on the Information System, and take out the record to send to the *SQL Query Generator Module*.

Table 1. SPARQL Query

```
PREFIX owl: < http://www.owl-ontologies.com/Ontology1233754688.owl#>
SELECT ?x ?y
WHERE
{
  ?x a owl:className.
  ?x owl:Nome_Database ?y.
}
```

In practice, this module receives in input a message containing:

- Shared Ontology URL
- Class where we could search instances and related DataProperty

and it creates a SPARQL query (Table 1) that get out for a given Shared Ontology Class, a matrix:

Table 2. Matrix of Database Pairs

| <i>TableName</i> | <i>DatabaseName</i> |
|------------------|---------------------|
| User | StaffDB |
| Degree | UniversityDB |

The matrix pairs (Table 2) will be useful to the SQL Query Generator to understand what is the table inside the database (pointed in the field DatabaseName) , and that correspond to the ontology class represented in the field TableName. This operation is possible thanks some mapping table created over the databases during the mapping procedure. It's possible that exists more instances, because data can be located in different table of different databases.

3.3 Mapping Layer

Here is defined an ontology that could represent different data coming from different databases, in the homogeneous way over an Information System.

Ontology db mapping module

Ontology db mapping is a fundamental module inside the mapping component, and it's charged to obtain an ontology from a company database. The obtained ontology will be merged in the upper layer (ontology layer that contains the shared ontology), in order to link the Information Systems with the outer world: data required by other systems will be retrieved thanks to those ontology help. Ontology must use meaning planned words, because it's very important to maintain shared syntactic basis. For this reason are used an updatable vocabulary and/or a domain ontology created by a domain expert, and located into the vocabulary box component.

The application created to accomplish the module task, allows to map in semi-automatic fashion, any database over an ontology. In order to do that, it uses some pre-established mapping criteria: tables are changed in class or subclass of generated ontology, relations in Object properties and attributes in dataproperties.

During the mapping procedures, every word are searched in WordNet (or in a domain ontology) before to be mapped[21]. If a word is not found, the user should change it with a word contained in WordNet (or in a domain ontology) or something else he should add this terms to the domain ontology.

Therefore, even if the original name of mapped classes could be changed by the user, the reference to them are stored into particular mapping tables creating by the application during mapping phase. It doesn't exists high performance tool to make a data research over an ontology, for this reason databases instances are stored only into databases and not into the created ontologies. SQL Manager is another fundamental module that will be explained later.

SQL Query Generator

The SQL Query Generator module, is located inside the mapping component, and its task is to retrieve data inside the singles agents databases, and to send the results to the communication systems. This module receive from the SPARQL Module, an input message containing:

- n pair tableName-databaseName where can exists data that we are looking for (sent by SPARQL Query Generator Module);
- data searched;
- some additional condition,

and send in output over the communication system, the results of the query encapsulated into an ACL message.

The SQL Query Generator, must interact with each Information System database, to analyze the relation between mapped database and derived ontologies, and to generate the query that should return data searched.

According the mapping rules used during the generation of local ontology from databases, each relational database table is mapping to a class in the related ontology, and each column in a DataProperty. Therefore the first value of n pairs sent by SPARQL module represent a possible relational model table, while the searched data is a table attribute. Some class name or attribute could not have corresponding element on the related mapped database, because they are changed by the user during the mapping process, however the mapping application create into the database the aid tables that stores correspondence by tables and classes names.

The first step of methodology used for the sql query generation, is the rescue of *ClassName-TableName* and *DatatypePropertyName-AttributeName* correspondence. The query format will be:

SELECT attribute FROM table WHERE Array[0][0]=Array[0][1] AND Array[1][0]=Array[1][1]

Where *attribute* represents the searched data when it's on the selected database, and *table* represents the table that corresponds to the class sent in input to the module, and *Array[i][i]* is an array containing all the attributes and values used to search data. For instance in a query must search the SSN of John Doe, the query will be:

SELECT ssn FROM user WHERE Name=John AND Surname=Doe.

The query result will be sent to the communication system.

3.4 The Communication System

The Communication System task, is to forward the information requests (and responses) sent by all Information Systems in its own domain. In order to do that, since in a single domain can coexists many Information Systems, it's useful a method that can allows to forward the requests only to the Information System that can have the required data. To resolve this problem, an internal mechanism inside the Information System creates a Global Ontology by applying a particular merging algorithm to the Shared Ontology sent by the Information System. This ontology represents a sort of semantic content index that allows to do a controlled request flooding. If the Information Systems can be seen as an agents, we can use a multi-agent system that is composed by multiple interacting intelligent agents. JADE is the multi-agent system selected, because it also allows to handle ontology both on server and agents. The communication system contains the *Interpreter* Module and the *Ontology Box* component.

The Interpreter receives the Finder request in input, consult the Global Ontology and forward these requests to all agent that contains a reference to the research criteria on the Global Ontology.

4. PROPOSED ONTOLOGY-MATCHING METHODOLOGY

On the basis of the architecture shown in figure 3, this paper proposes a methodology that, following the approach of ontology-matching, allows to generate an Information System shared ontology that represents the semantics of the information contained in the Information System data schema. Regarding the architecture shown in figure 3, methodology places both inside the ontology layer and inside the communication system.

To describe how methodology works, we use two ontology (figures 4 and 5) derived from two different EER models that represent in two different way the university domain.

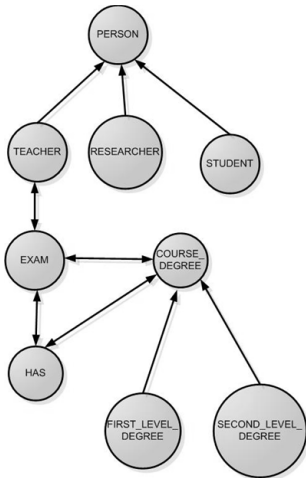


Figure 4: First ontological model

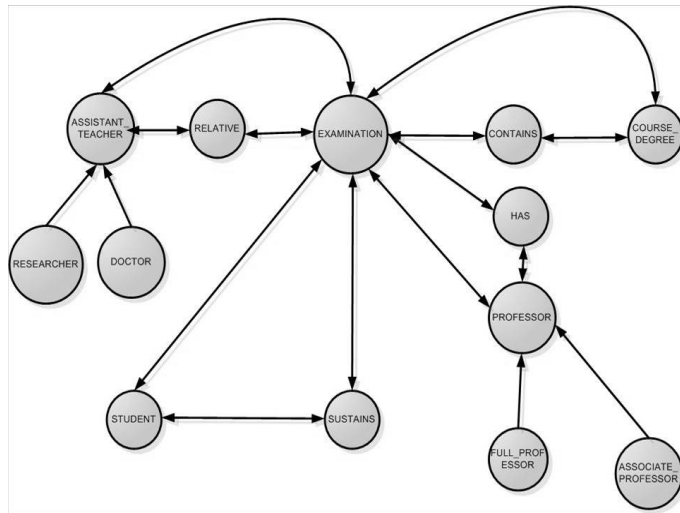


Figure 5: Second ontological model

The proposed ontology-matching methodology allows to obtain a shared ontology that defines, beginning from the ontologies obtained starting from the database, semantically equivalent classes.

In order to make effective the methodology for the shared ontology generation, through the ontology-matching approach, we need to considerate three main aspects about the semantic interpretation of the information: the *information representation*; the *semantic disambiguation of the information*; the *semantic similarity of the information*. Methodology is composed, therefore, of the phases: *linguistic normalization*; *semantic disambiguation*; *semantic similarity*; *tree matching* (figure 6). The problem of the information representation (faced in the linguistic normalization phase) comes from the adoption of different morphological forms in the definition of the data presents in a model; moreover, in every information they are often adopted special characters that must be excluded during the application of the methodology because they do not own some semantic value.

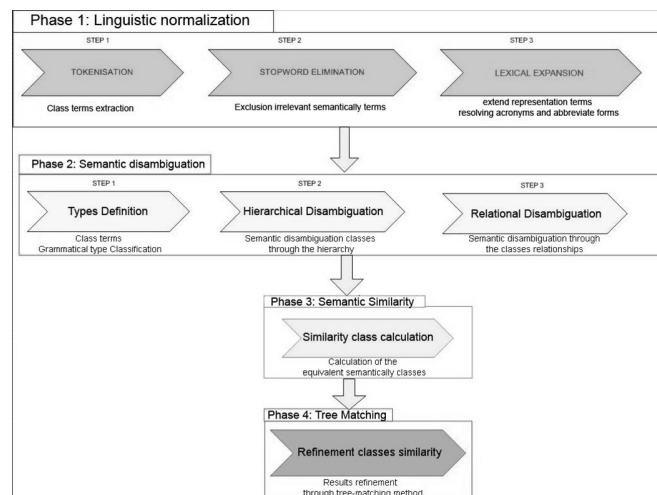


Figure 6: Phases of the ontology-matching methodology

The problem of the disambiguation and semantic similarity is dealt using WordNet that provides a tool to solve both the disambiguation and the semantic distance among two objects. Such problem has been described in the second phase of the methodology, semantic disambiguation, in which will be tried to establish, beginning from the classes of every ontologies, the real meaning (sense) for each of them. Once established the sense of each term in the ontology, it will be determined, in the third phase of the methodology (semantic similarity), the semantic similarity degree among each of them so that to be able to seek equivalent classes. The last phase of the methodology is the refinement phase (Tree matching) that it modifies the similarity degree among two classes in relationship to the equivalences of the hierarchies. In the following paragraphs are detailed the methodology phases shown in figure 6.

4.1 Linguistic Normalization

In the linguistic normalization phase, the goal is to represent, in the same form, all the classes of the ontology. To conform to the representation of each class is possible to resolve three types of differentiations: Morphological, Syntactic and Semantic. The use of WordNet, in the methodology, allows to obtain, for any word, the word corresponding to the concept that represents; for example, both words *works* and *working* are considered equivalent to the word *work* of which they constitute a grammatical variation. Moreover, WordNet allows, in semantic similarity calculation area, to execute a semantic normalization of the words in the ontology. Instead, it is important to face, in this phase, three aspects that allow to represent, in a consistent semantically way, every single class:

- **Tokenisation.** This operation allows to represent each class like a set of terms. To do so, it is necessary to exclude the characters without meaning as so punctuation marks or special characters. Beginning from the exclusion of these characters, each class will be represented as a set of terms. For example, the COURSE_DEGREE class will be equal to the union of the terms COURSE and DEGREE: COURSE_DEGREE = (COURSE,DEGREE)
- **Stopword elimination.** Beginning from the terms derived from the tokenisation phase, they are excluded terms that don't have semantic value for the ontology-matching goal. We have decided that the terms to be excluded are the articles and the prepositions.
- **Expansion.** Often in the database it is possible to find shorten forms (for example CodFisc for Fiscal Code) or acronyms (UoM for Unit of Measure). The methodology foresees the use, optional, of a thesaurus that "expands" such representations.

It is important to notice as, at the end of the normalization phase, we will have a set of representation (figure 7) of each class that will be constituted by the union of the corresponding terms (semantically relevant).

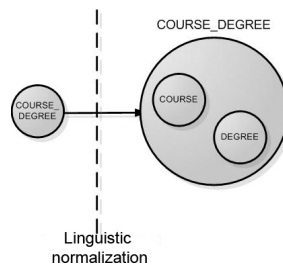


Figure 7: Linguistic normalization phase

Beginning from this phase, it will be possible, considering each class term, to obtain the semantic class meaning and the similarity degree with the other classes.

4.2 Semantic disambiguation

In the semantic disambiguation phase it establishes, for each class term, the corresponding sense in WordNet. For example, if we consider the word *course*, WordNet will connect this word to more than 10 senses that define as many different concepts expressed by this word and represented through this type notation:

WORD#SENSE_NUMBER

For example, considering the COURSE_DEGREE class, the semantic disambiguation phase should get for the terms COURSE and DEGREE the following senses:

course#1 : (education imparted in a series of lessons or meetings) *"he took a course in basket weaving"; "flirting is not unknown in college classes";*
degree#3 : (an award conferred by a college or university signifying that the recipient has satisfactorily completed a course of study) *"he earned his degree at Princeton summa cum laude";*

This phase of the methodology is composed of these steps:

Types Definition: according to the term types that compose each class, a different metrics will be applied for determining the sense of each of them. Particularly, a first operation will be performed, in which the methodology will have to establish, with the user collaboration, the typology of the terms that constitute each class (classes, verb, agentive or adverbs). From different experimental tests executed on WordNet, it has emerged, for a word, the frequent presence of different senses for different grammatical "types". The system will evaluate the probabilistic value of grammatical types for each word: for a class constituted by only one word, it is considered as more probable than the associate term is a noun or a verb. Therefore, in case that a word that represents a class has different corresponding grammatical types, the system will evaluate with greater probability the affiliation to the names or the verbs class. In a class constituted from more words, it increases the probability that a term is: an adjective, in presence of a noun inside the class; a noun, in presence of a verb inside the class. The results achieved by the system, require, for the cases of ambiguity the possibility of user validation. This methodology face the semantic disambiguation beginning from two different levels of inter classes analysis, in which the sense of a term is evaluated in a class in relationship to the characteristics of the context (connected classes) and intra classes, in which the sense of a term is evaluated in relationship to the concept expressed by the other terms in the class.

Hierarchical Disambiguation in this step we consider the nouns present in the child class, then will be determined the sense performing a similarity calculation based on the Leacock-Chodorow metrics. Particularly, according to the intuition and the analysis made up by Warin, Oxhammar, & Volk [22], the sense of a noun will be determined in the child class calculating the sum among the similarity degree with the all possible senses of the father class. Of course, if at the end of the types definition phase, the noun of the father class will already have an assigned sense, the nouns of the child class will be compared only with the noun of the father class. The metrics, doesn't determine the sense associate with the root classes in a hierarchy. We have decided, therefore, in the methodology, to set the nouns sense of these classes, beginning from the sense, determined in the previous step, of the child class. To do this, we use the Wu-Palmer metrics [23], because it is more appropriate in so far determining the most correct sense exclusively not founding on the distance. We consider, now, an example of application of the step of hierarchical disambiguation. For example, if we consider the ontology in figure 5, at the end of the types definition step for this ontology, the classes ASSISTANT, DOCTOR and RESEARCHER will have the characteristics shown figure 8.

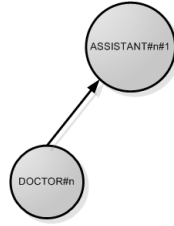


Figure 8: Hierarchical disambiguation

In the class doctor, the corresponding term haven't an associated sense. Applying the Leacock-Chodorow metrics, and comparing doctor with assistant#n#1, we obtain the results shown in figure 9:

| Measure | Word 1 | Word 2 | Score |
|---------|------------|---------------|--------|
| lch | doctor#n#4 | assistant#n#1 | 1.8971 |
| lch | doctor#n#2 | assistant#n#1 | 1.743 |
| lch | doctor#n#1 | assistant#n#1 | 1.6094 |
| lch | doctor#n#3 | assistant#n#1 | 1.0498 |

Figure 9: Leacock-Chodorow metric results

The determined sense, individuated according to the high score, it is correct (doctor#n#4), because we have in WordNet the following definition:

doctor#4: (a person who holds Ph.D. degree (or the equivalent) from an academic institution) *"she is a doctor of philosophy in physics"*

Relational Disambiguation: it considers the disambiguation relatively to those root classes in the ontology, that haven't subclass and for which it is not possible, therefore, to use the hierarchical disambiguation step. This phase operates in two steps in relation to the terms type present in each class: Nouns and Verbs: for this terms type, the sense evaluation is made up according to the semantic relationship with the near classes (inter classes); Adjectives and adverbs: for these term types the evaluation of the sense is realized according to the semantic relationship between the nouns present in the same class. In this phase, we need to evaluate the semantic relationship level to be able to establish, in a correct way, the sense associate to a term. In this area, drawn on the analysis of Patwardhan & Pedersen [24], it obtains that the Gloss Vector metrics is more effective. In the first step of inter classes disambiguation between nouns and verbs, it is necessary to determine, beginning from a class, on which classes we achieve the semantic comparison. Achieved the connected classes, it will pass to compare with one of such classes following this principle: the class has to be select, among those having a sense already determined; if there aren't connected classes, with a sense already assigned, we select the class with the smaller number of terms(it is easier to compare); if the class haven't classes directly connected, through object property, it goes up to the connected classes through the classes that represent the M-N relationships; for example, from the class EXAMINATION can be gone up to both the STUDENT class and the ASSISTANT class. We consider now, for example, the class EXAMINATION; this class is directly connected, through object property, to the PROFESSOR class, whose sense is unique and already determined (professor#n#1); we can apply, now, the previous procedure achieving the results in figure 10:

| Measure | Word 1 | Word 2 | Score |
|---------|----------------|--------------|--------|
| vector | examination##2 | professor##1 | 0.1894 |
| vector | examination##3 | professor##1 | 0.1273 |
| vector | examination##1 | professor##1 | 0.1205 |
| vector | examination##5 | professor##1 | 0.0768 |
| vector | examination##4 | professor##1 | 0.049 |

Figure 10: Gloss-Vector metric results

The sense achieved by the metrics, individuated according to the high score, therefore, is correctly: examination##2; to this sense corresponds to the following definition :

examination##2: (a set of questions or exercises evaluating skill or knowledge) *"when the test was stolen the professor had to make a new set of questions"*

4.3 Semantic Similarity

After the methodology semantic disambiguation phase, we compare among classes belonging to the different ontologies using the two followings metrics:

- Wu-Palmer for the comparison among a couple of noun;
- Gloss Vector if at least term of couple is a verb, an adjective or an adverb.

Within the methodology each class is constituted by a set of terms; to determine the general value of similarity it is necessary to calculate an average of the similarity value obtained for each term in the set. Particularly, a class term will be compared with all terms belonging to the other class; its similarity value will be given from the maximum one among the obtained ones. We suppose, for example, to have two classes T1 and T2 constituted by a set of terms, the similarity degree will be obtained through the following formula :

$$ns(T_1, T_2) = \frac{\sum_{t_1 \in T_1} \left[\max_{t_2 \in T_2} sim(t_1, t_2) \right] + \sum_{t_2 \in T_2} \left[\max_{t_1 \in T_1} sim(t_1, t_2) \right]}{|T_1| + |T_2|}$$

Figure 11: Similarity among two classes calculation Formula

In the formula in Figure 11, t1 represents a term present in the T1class, while t2 represents a term present in the T2 class; |T1| and |T2| represent, respectively, the number of terms present in the class T1 and the number of terms in the class T2. We suppose, for example, to want to determine the similarity degree between ASSOCIATE_PROFESSOR (figure 5) in one ontology and TEACHER in the other one (figure 4), that have the following senses:

ASSOCIATE_PROFESSOR : associate#a#1 , professor##1

TEACHER : teacher##1

In the semantic similarity phase, each class in an ontology will be compared, following the formula in figure 11. This choice is due to the global nature of the matching operation, that requires to determine among all the classes present in all ontologies those that have a greater equivalence degree. The result obtained will be in the case two ontologies in tabular form shown in figure 12. In the evaluation of the equivalence between two classes we decided to consider a least value of threshold equal to 0,6.

| | PERSON | TEACHER | RESEARCHER |
|----------------|--------|---------|------------|
| ASSISTANT | 0,833 | 0,625 | 0,7143 |
| RES.DOC | 0,46 | | 0,7143 |
| RESEARCHER | 0,2530 | | 1 |
| PROFESSOR | 0,6667 | 0,88 | 0,5882 |
| FULL_PROFESSOR | 0,59 | 0,647 | |
| EXAMINATION | | | |
| STUDENT | | | |
| COURSE_DEGREE | | | |

Figure 12: Methodology result example

4.4 Tree Matching

The last phase of the matching algorithm represents a phase of refinement in which, proceeding from the leaf node up to the root node of the involved ontologies, it checks if the father classes has or not a similarity degree greater to a determined threshold value, imposed by the methodology (Cshold); in the positive case the similarity degree of the leaf classes will be increased of one determined constant percentage (Cinc), otherwise the similarity degree will be decrease of another constant percentage (Cdec). After several experimental analyses on the methodology the values selected for Cshold and Cdec are:

- $C_{sold} = 0,6$
- $C_{dec} = 0,1$

These values may be updated.

4.5 Generation of the shared ontology

From the methodology of matching previously described, beginning from the semantic similarity results, we obtain the shared ontology that describes the Information System semantic domain. In the generation of the shared ontology it represents in only one class all the classes that are semantically equivalent and that are distributed in the domain ontologies; this class, therefore, will represent an useful tool, during a semantic integration scenario, to understand the type of information present in an Information System, and where they are positioned.

In figure 13 there is a portion of shared ontology obtained in an experimentation phase starting from the two ontologies in figure 4 and figure 5. Methodology has obtained the equivalence between the class RESEARCH_DOCTOR and RESEARCHER, between the classes PROFESSOR and TEACHER and between the classes STUDENT.

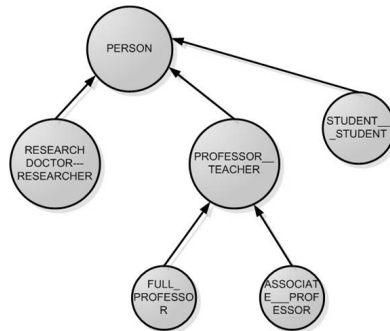


Figure 13: Ontology shared model derived from methodology

5 CONCLUSIONS

This focus of the paper is on the semantic integration between Information System. The paper present an architecture of integration between information systems that, starting from the

database of each companies' information systems, allows to extract the ontology that represents the information. The architecture has a component that allows to merge ontologies both in each information system (when it is made up of several databases) –creating the shared ontology-and between several information systems-creating the global ontology.

The paper focus on this component and proposes a methodology that, starting from already existing algorithms, drive in the realization both of the shared and of the global ontology . The output of the application of the methodology is a semantic net in which are semantically represented the equivalent information that is in the data sources distributed in an Information System. Through the relationships present among the classes of the ontology, it is possible to go up to all information requested in a scenario of semantic integration. Our next step in this research work will be to make some experiments of the application of the methodology in a real Information System in order to verify the degree of precision of our work. In order to do so, we are designing and implementing a tool to support in the automatic/manually application of the methodology here proposed.

REFERENCES

- [1] Dittrich, P. Z. (2004). "Three decades of data Integration." *In 18th IFIP World Computer Congress*.
- [2] Gruber, T. R. (1993) "A Translation Approach to portable Ontology Specifications." *Knowledge Acquisition. Knowledge Acquisition Vol 5*.
- [3] Guha, L. (1990). *Building large knowledge-based systems*. Addison Wesley, Reading.
- [4] Farrar, S (2003) "A linguistic ontology for the semantic web". *GLOT International*.
- [5] Zhang S., Bodenreider O., & Golbreich C., (2006) "Experience in Reasoning with the Foundational Model of Anatomy in OWL DL." *Pacific Symposium on Biocomputing*.
- [6] Jérôme Euzenat, P. S. (2007) *Ontology Matching*. Springer.
- [7] Rybinski (1987). *On First-order-logic databases*. *ACM Transaction on Database Systems*.
- [8] Baader, F. C.-S. (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge : Cambridge University Press.
- [9] An, Y. B. (2005). "Inferring complex semantic mappings between relational tables and ontologies from simple correspondences". *In Proceedings of International Conference on Ontologies, Databases and Applications of Semantic pp*. 1152-1169
- [10] Bizer, C. (2003). *D2R MAP – A Database to RDF Mapping Language*. Budapest, Hungary.
- [11] Jesús Barrasa, O. C.P. (2004). *Fund Finder: A case study of database-to-ontology mapping*. Madrid, Espana.
- [12] Trinh, Q. Et al, 2007. Semantic Interoperability Between Relational Database Systems. *Proceeding in the 11th International database engineering and application symposium IEEE pp*. 208-215.
- [13] OASIS Service Component Architecture / Assembly (SCA-Assembly) T.C., *Service Component Architecture Assembly Specification Version 1.0*, March 2007.
- [14] OASIS SOA Reference Model TC., *OASIS Reference Model for Service Oriented Architecture*, April 2008.
- [15] JADE. 2009. Jade official homepage. <http://jade.tilab.com>: <http://jade.tilab.com/>
- [16] FIPA. 2009. FIPA official homepage <http://www.fipa.com>
- [17] DDS reference, *Data Distribution Service for real-time systems*, January 2007
- [18] Paiano, R., Guido, A.L. (2009). "Semantic data integration: overall architecture" *International Business Information Management Conference (11th IBIMA) 4 – 6 January 2009 Cairo, Egypt*
- [19] W3C February, 10 2004. OWL Web Ontology language Reference (W3C)

- [20] Guido, A. and Paiano R. 2009. *Semantic Integration Between Information Systems by Ontology*. Lecce, Italy
- [21] Warin, M. Et al. 2005 *Enriching an Ontology with WordNet based on Similarity*. Stockholm, Sweden.
- [22] Warin, M., Oxammar, H., & Volok, M. (2005) "Enriching an Ontology with WordNet based on Similarity". *In Proceedings of the MEANING-2005 Workshop*.
- [23] Zhibiao Wu and Martha Palmer (1994) "Verb semantics and lexical selection." *In 32nd Annual Meeting of the Association for Computational Linguistics*.
- [24] Patwardhan, S., & Pedersen, T. (2006) "Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts". *In Proceedings of the EACL 2006 Workshop*.