SEMANTIC INTEGRATION FOR AUTOMATIC ONTOLOGY MAPPING

Siham AMROUCH[1] and Sihem MOSTEFAI[2]

[1]Computer Science Departement, Med Cherif Messadia University,
Souk Ahras, Algeria
sihamamrouch@yahoo.fr
[2]MISC Laboratory, Computer Science Departement, Mentoury University
Constantine, Algeria
xmostefai@yahoo.com

ABSTRACT

In the last decade, ontologies have played a key technology role for information sharing and agents interoperability in different application domains. In semantic web domain, ontologies are efficiently used to face the great challenge of representing the semantics of data, in order to bring the actual web to its full power and hence, achieve its objective. However, using ontologies as common and shared vocabularies requires a certain degree of interoperability between them. To confront this requirement, mapping ontologies is a solution that is not to be avoided. In deed, ontology mapping build a meta layer that allows different applications and information systems to access and share their informations, of course, after resolving the different forms of syntactic, semantic and lexical mismatches. In the contribution presented in this paper, we have integrated the semantic aspect based on an external lexical resource, wordNet, to design a new algorithm for fully automatic ontology mapping. This fully automatic character features the main difference of our contribution with regards to the most of the existing semi-automatic algorithms of ontology mapping, such as Chimaera, Prompt, Onion, Glue, etc. To better enhance the performances of our algorithm, the mapping discovery stage is based on the combination of two sub-modules. The former analysis the concept's names and the later analysis their properties. Each one of these two sub-modules is it self based on the combination of lexical and semantic similarity measures.

KEYWORDS

Automatic ontology mapping, semantic integration, wordNet, owl, Lexico-semantic similarity.

1. INTRODUCTION

In the last decade, the aim of researchers in semantic web community was and still to bring the actual web to its full potential by considering ontologies as the best means to annotate the data on the web [7]. The well known and the most referenced definition of ontology is that of T. Gruber in 93, that defines an ontology as an explicit specification of a conceptualization [1]. This structure can be cognitively semantic (ontology intended to be exploited by the user) or computationally semantic (ontology intended to be exploited by the machine), [2]. We may define an ontology as a taxonomy of classes related with a set of hypo(hyper)nym relationships, where each sub-class

Sundarapandian et al. (Eds) : ICAITA, SAI, SEAS, CDKP, CMCA-2013
pp. 387–396, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3832

describes a concept that is more specific than the concept described by the super-class. As more and more ontology developers get involved, many ontologies are created for describing similar or even complementary domains. Interoperability among different ontologies becomes essential to take advantage from the semantic web. It is required for combining distributed and heterogeneous ontologies. Hence, to promote the interoperability between these differently designed ontologies, ontology mapping is a prominent solution. They enable people and software agents to work in a more smooth and collaborative way [3]. In deed ontology mapping have a significant effect on promoting automatic interoperability. They also, play motivating roles for developing ontologies by reusing existing open ones and integrating ontology based web data sources, which reduces the costs of ontology engineering and remotes the use of standard tested ontology modules [17]. While ontology mapping describes a set of correspondences between correspondent concepts through two (or more) ontologies, ontology alignment is the process that takes two input ontologies and produces a set of relationships between concepts that matches semantically with each other . These matches are called "Mappings" [5]. The semantic matches described by the mappings can denote relations of equivalence (is-a), specialization and/or generalization (part of), as they may indicate other senses. Mappings can solve different forms of mismatches [20] (i) Language level mismatches or syntactic mismatches, caused by using different ontology representation languages and (ii) ontology level mismatches or semantic mismatches such as synonyms, homonyms, hyponyms, etc. Several tools and algorithms for ontology mapping and alignment exist on the literature. Most of them are semi-automatic and require a lot of human intervention to resolve the different forms of mismatches because their algorithms do not use the semantics embedded within ontologies. To avoid these insufficiencies, we have integrated semantics to propose a new algorithm for fully automated ontology mapping that creates semantic bridges between similar concepts belonging to different source ontologies. In this paper, we present in detail, how linguistic analysis is applied on the names and properties of concepts to scale out the similar ones. The similar concepts will be mapped to each other in the resulting bridge ontology.

This paper is organized as follows: Section 2 briefly describes the ontology interoperability techniques. Sections 3 and 4 outline successively the semantic integration and mapping discovery processes. Section 5 surveys the literature of related works. Section 6 presents the architecture of the general proposed algorithm. In section 7 we compare our proposed algorithm with the well known existing algorithms and we conclude by stating some important remarks and possible prospects in Section 8.

2. SEMANTIC INTEGRATION

This is a substantial research field that serves the semantic web by facilitating interoperability between different applications and/or knowledge sources such as ontologies. The semantic integration or even enrichment is performed through external resources such as domain specific dictionaries. An example of the most known and general designed computerized dictionary is WordNet[1]. We recall that WordNet is a computerized english dictionary where the basic unit is the concept. It uses two different means to define the meaning of a word, the synsets and the lexical relations. A word is then defined by a set of synonyms (synset) and a definition. Example: Board: synset = {board, blank} Definition: A piece of wood.

[1] http://www.wordNet.princeton.edu/wordNet

These external resources are used to avoid the limitations of the lexical aspect in the ontology mapping process after their possible extensions according to their application domain. So, it is at this stage where the semantic aspect acts to support the mapping discovery process. Herein, the more the extension of the source ontologies is close to the same shared ontology, the easier will be the mapping identification process. In addition, reasoning and inference processes allowed by the ontology representation languages contribute in specifying the constraints of similar concepts mapping.

3. MAPPING DISCOVERY

It's a common sub-process for the three processes of ontology Mapping, Aligning and Merging. Its objective is to identify similar concepts in source ontologies. The concepts judged similar will be matched by mapping relations (when Mapping or Aligning ontologies) or merged into a single concept (when Merging ontologies).

We recall that an ontology is designed and developed to serve as a common vocabulary that is shared by multiple applications and communities of information system developers, which is not possible with data bases. The reader can refer to [21] for more details on the limitations of data bases that may be avoided by ontologies. A common ontology is then accomplished by domain or application-specific concepts and properties by the knowledge engineer. The mapping discovery process can be effectively very easier if it is between two extensions that refer to the same common ontology. In addition, ontologies are developed to be manipulated by inference engines. And ontology representation languages are specified at the basis of reasoning. Then inference and reasoning have a prominent effect to discover mappings between the two ontologies under discussion. Based on these two aspects, Noy [4] scales two major architectures to find mappings between source ontologies:

Using a shared ontology, where the common ontology is accomplished by the application-specific concepts and properties. The More these extensions are consistent with the definitions provided in the common ontology, the easier will be the mapping discovery between the two extensions. **Using heuristics and machine learning.** Herein, lexical and structural components of definitions are used. They exploit the semantics contained in ontologies (semantics of relations: is-a, part-of, attachment of properties to classes, property domain and co-domain definitions, etc.). In contrast with data base schemas, ontologies have much more specific constraints. These ones provide main basis for the automatic mapping (matching) discovery methods.

4. RELATED WORK

Several tools for ontology Mapping or Alignment and even Merging exist in the literature. Most of these tools are semi-automatic and the design of fully automatic tools is usually a delicate issue. In this section, we outline the well known and recent ones:

FCA-Merge [12]. It's a method for semi-automatic ontology merging. Its process is summarized as follows: First, from a set of input documents, popular ontologies (ontologies equipped by their instances) are extracted. Once the instances are extracted and the concept lattice is constructed, FCA-techniques are used to generate the formal context of each ontology. Using lexical analysis, FCA-techniques retreive specific information that combines a word or an expression to a concept if it has a similar concept in the other ontology. Then the two formal contexts are merged to

generate the pruned concept lattice. Herein, the knowledge engineer may eventually intervene to resolve conflicts and eliminate duplications using his background about the domain. It should be mentioned that the major drawback of FCA-Merge is that it is based on instances to identify similar concepts, however, in most applications, there are no objects that are simultaneously instances in both source ontologies.

PROMPT [13] is an interactive ontology merging tool, it proposes a list of all possible merging actions (to-do list). After that, the knowledge engineer selects the appropriate proposals that go with his needs. Then, PROMPT automatically merges the selected pairs of concepts, provides the conflicts generated after merging (conflict-list) and proposes their appropriate solutions. Finally, the knowledge engineer selects the most suitable solutions.

CHIMAERA [14]. An interactive ontology merging tool, where the knowledge engineer is charged to make decisions that will affect the merging process. Chimaera analyzes the source ontologies and if it finds linguistic matches the Merging is performed automatically, otherwise, the user is prompted for further action. Like PROMPT, Chimaera is an ontology editor plugin, namely Ontolingua, but they differ in the suggestions they make to their users with regard to the merging steps.

GLUE [15]. To find mappings between two source ontologies O' and O'', Glue uses machine learning techniques. So, for each concept of ontology O', Glue finds its most similar concept in ontology O" based on different practical similarity measures and several machine learning strategies. The authors also used a technique called "relaxation labeling" to map the two hierarchies of the two ontologies. This technique assigns a label to each node of a graph and uses a set of domain independent constraints, such as, two nodes of concepts c' and c" match if the nodes of their neighbourhood2 v(c') and v(c") also match, and a set of domain dependent constraints, such as, if X is an ascendent of Y and Y matches "direction" then X does not match "sub-direction".

ONION [16]. According to the authors, ontology Merging is inefficient because it is costly and not scalable. So, ONtology compositION system provides an articulation generator for resolving mismatches between different ontologies. The rules in the articulation generator express the relationship between two (or more) concepts belonging to the ontologies. Manual establishment of these rules is a very expensive and laborious task. And full automation is not feasible due to the inadequacy of natural language processing technology. The authors also elaborate on a generic relation for heuristic matches: Match gives a coarse relatedness measure and it is upon to the human expert to then refine it to something more semantic, if such refinement is required by the application. In their system, and after validating the suggested matches by a human domain expert, a learning component is included in the system which uses the user's feedback to generate better articulation in the future when articulating similar ontologies.

5. GENERAL PROPOSED SYSTEM

The aim of our work is to propose a new algorithm for fully automatic ontology Mapping system. First, we import the two source ontologies that cover the same or complementary application domains. Next, we identify their similar concepts. Here, we will use an Information Retrieval (IR)

[2] neighbourhood is defined to be the children, the parents or both.

technique, where each concept of the first ontology is compared with all concepts of the second one. To avoid a lot of unnecessary comparisons, we begin from the top of one ontology and from the bottom of the second one. We will base our method on linguistic analysis of concepts' names to compute the lexical and semantic similarities between them. But, sometimes, we may find two similar concepts that are described by the same or similar properties but which are labelled with different strings, for example, one concept can be labelled with the abbreviation of its name or even by a code, in such case, it is impossible to identify its synonyms from the used dictionaries. To resolve this problem, we have combined this module with another module that is based on the linguistic analysis of concept's properties. Both sub-modules of the similarity identification module combines the results of two similarity measure techniques used in string comparison (concepts or properties), one of them is lexical and the other one semantic. After that, the concepts accepted as similar are mapped to each other through the relation "is-a". Hence, the resulting bridge ontology is constructed and the ontology interoperability is handled through the resulting semantic bridge.

A. LEXICAL SIMILARITY

This technique is based on the computation of a distance between two strings describing the names of two concepts. Several measures of similarity or distances exist in the literature such as Levenstein distance [8], Hamming distance [11], Jaro distance [9], Jaro-winker distance [10], etc. All of these measures are based on the same assumption described by [6] which states that two strings are similar if they share enough important elements. We have chosen to use the Jaro distance as a similarity measure because it yields a value which is consistent with the value given by the semantic similarity measure that we have proposed (a value between 0 and 1) and therefore their combination is easier. The lexical similarity between the two concepts c1 and c2 is given by:

$$SIMlex(c1, c2) = Dj(s1, s2) \qquad (1)$$

where Dj(s1,s2) is the Jaro distance between the two strings s1 and s2 labelling the two concepts c1 and c2 and which is defined by the equation :

$$Dj(s1, s2) = \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m - t}{m} \right) \qquad (2)$$

Where: m: The number of matched characters. t = N / 2: the number of transpositions.

N: The number of pairs of matched characters that are not in the same order in their respective chains. Two identical characters of s1 and s2 (describing the concepts c1 and c2 respectively) are considered matched if their distance (i.e. the difference between their positions in their respective chains) does not exceed a certain value given by:

$$val = \left[\frac{\max(|s1|, |s2|)}{2} \right] - 1 \qquad (3)$$

The two concepts c1 and c2 are considered lexically similar if the distance between them exceeds a critical threshold to be determined empirically.

Example: Computation of lexical similarity between 'auto and automobile' and between 'auto and car':

SIMlex (auto, automobile) = Dj(auto, automobile)= ?

	a	u	t	o	m	o	b	i	l	e
a	1	0	0	0	0	0	0	0	0	0
u	0	1	0	0	0	0	0	0	0	0
t	0	0	1	0	0	0	0	0	0	0
o	0	0	0	1	0	1	0	0	0	0

m=5 (number of 1 in the table), |s1|=10, |s2|=4, N=1, t=1/2,

$$SIMlex(auto, automobile) = \frac{1}{3}\left[\frac{5}{10} + \frac{5}{4} + \frac{5-0.5}{5}\right] = 0.883.$$

Assuming that the threshold = 0.5, SIMlex = 0.883 ≥ 0.5 then auto and automobile are lexically similar.

Now, let's compare "car" and "auto", SIMlex (car, auto) = Dj(car, auto)= ?

	a	u	t	o
c	0	0	0	0
a	1	0	0	0
r	0	0	0	0

m=1, |s1|=4, |s2|=3, N=1, t=1/2, $SIMlex(auto, car) = \frac{1}{3}\left[\frac{1}{4} + \frac{1}{3} + \frac{1-0.5}{1}\right] = 0.36.$

SIMlex(auto, car) = 0.36 <0.5, then auto and car are lexically dissimilar.

SIMlex(car, plane)=0.34 <0.5, then plane and car are lexically dissimilar.

B. SEMANTIC SIMILARITY

When the concepts are semantically similar but their names are different (synonyms) the null lexical similarity does not reflect the reality. To solve this problem, the integration of semantic similarity measure is crucial. To do this, we have begun with a semantic enrichment of the two source ontologies from wordNet. It involves building a synonymy vector containing the synset elements for each concept.

For the computation of semantic similarity, we have used an information retrieval technique, which involves comparing each concept in the first ontology with all concepts of the second one to find out the most similar concept. We defined the semantic similarity between two concepts C1 and C2 as follows:

$$SIMsem\ (c1,c2) = 2 * \frac{card\ (synset\ (c1) \cap synset\ (c2))}{card\ (synset\ (c1)) + card\ (synset\ (c2))} \qquad (4)$$

SIMsem (c1, c2) Є [0,1].

The two concepts c1 and c2 are judged similar if *SIMsem (c1, c2)* is greater than a critical threshold which will be determined empirically. If the two concepts are exactly similar

SIMsem (c1, c2) =1, in the opposite case *SIMsem (c1, c2) =0.*

Example : Computation of lexical similarity between 'auto and car' and between 'car and plane':

Synset (auto)={car, auto, automobile, machine, motocar}, synset (car)={ car, auto, automobile, machine, motocar }, synset (plane)={airplane, aeroplane, plane}

$$\text{SIMsem(auto, car)} = 2 * \frac{5}{10} = 1. \quad \text{and} \quad \text{SIMsem(car, plane)} = 2 * \frac{0}{8} = 0.$$

Then auto and car are semantically similar but plane and car are semantically dissimilar.

Once the two similarity measures are computed, we compute the *lexico-semantic* similarity that combines the two results through the formula:

$$SIMlexSem(c1,c2) = \frac{SIMlex + 2 * SIMsem}{3} \qquad (5)$$

The two concepts are considered similar if *SIMlexSem (c1, c2)* reaches a critical threshold which will be determined empirically.

Example : $\text{SIMlexsem(auto, car)} = \frac{0.36 + 2 * 1}{3} = 0.75 > 0.5$ So, the two concepts auto and car are similar and then will be mapped to each other through the relationship "is-a" and hence, we obtain the semantic bridge "auto is-a car".

$\text{SIMlexsem(plane, car)} = \frac{0.34 + 2 * 0}{3} = 0.113 < 0.5$ So, the two concepts plane and car are dissimilar, so, they will not be mapped to each other in the resulting bridge ontology.

HOW THE BRIDGE ONTOLOGY IS CONSTRUCTED?

First, the two source ontologies are imported to an internal structure model based on owl. Then the module of mapping discovery is launched. Herein, each concept of the second source ontology is compared with all the concepts of the first one. Such that, we begin from the top of the first ontology and from the bottom of the second one. This technique avoids a lot of

unnecessary comparisons. This module is based on the combination of two sub-modules. In the first one, the lexico-semantic similarity is simply and directly computed on the strings naming or labelling the discussed concepts. Alone, this module may fail to discover all existing mappings. To overcome this limitation, we have combined it with another sub-module that is based on the similarity computation between their vectors of properties. Herein, each one of the two concepts is identified by its array of properties. Then each property of the first array is compared with all the properties of the second array, using usually the lexico-semantic similarity measure presented previously. Each time two properties are found similar, a counter *c* is augmented by one.

Finally, the ratio *R* of similarity between the two concepts (described by their properties) is computed through the formula 6, Where *p1* and *p2* are the arrays of properties of *c1* and *c2* respectively.

$$R = \frac{2 * c}{p1.length + p2.length} \qquad (6)$$

At the end, the two similarity measures (the one computed on concept's names and the other computed on concept's properties) are combined together by taking their average value. If this later reaches a critical threshold, that will be determined empirically, the two concepts are judged similar, and then, will be mapped to each other by the relation "is-a". This process is repeated for each concept of the second ontology. Hence, the whole bridge ontology is constructed. The whole proposed architecture of the fully-automated ontology mapping system is depicted by figure 1.

6. COMPARAISON WITH EXISTING ALGORITHMS

Finally, we compare the whole proposed algorithm with the most known ones that exist in the literature such as: CHIMAERA, ONION, PROMPT, FCA-MERGE and GLUE, throw a set of critical properties as shown on the table 1:

Table 1. Comparaison with existing algorithms.

Properties	CHIMAERA	ONION	PROMPT	FCA-MERGE	GLUE	PROPOSED
1 Automation	Semi-automatic	Semi-automatic	Semi-automatic	Semi-automatic	Semi-automatic	Fully-automatic
2 Operation	Merge	Composition	Mapping+merge	Merge	Mapping	Mapping
3 (In)dependence	Ontolingua	Independent	Protégé 2000	Independent	Independent	Independent
4 Representation langages	Ontologua	Labeled and oriented graphs + Horn rules	Rdfs – owl	Concepts taxonomies of populated ontologies	Taxonomies	Owl
5 External resources	No	WordNet	No	No	No	WordNet
6 Lexical matching	No	No	Yes	Yes	Yes	Yes
7 Semanticmatching	Yes	Yes	Yes	Yes	Yes	Yes
8 Instance matching	No	No	No	Yes	Yes	No
9 Structure matching	oui	No	Yes	Yes	Yes	No
10 User role	Takes decisions affecting the merging process	Validates the proposed mappings	Selects appropriate mappings from to-do list	Corrects conflicts and eliminates duplications	Selects the similarity computation function	No intervention

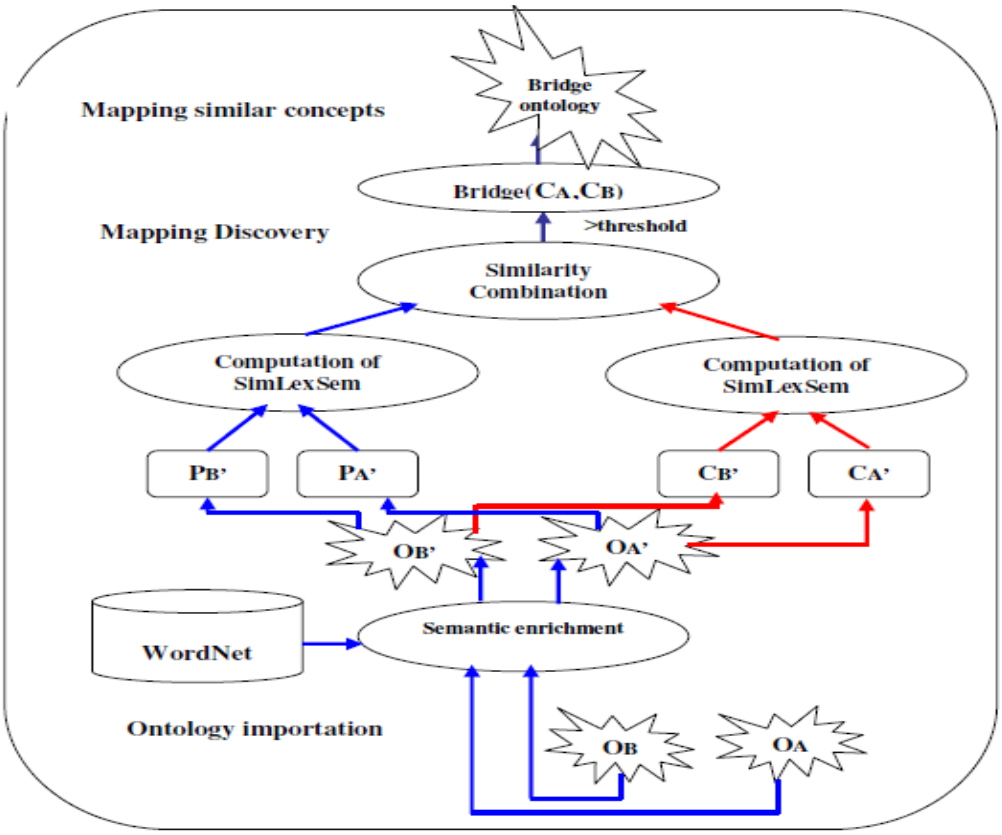

Figure 1. Proposed architecture for the fully-automatic ontology mapping system.

7. CONCLUSION

Ontology mapping process is a prominent technique to overcome the restrictions and specifications of information and knowledge when the application covers more than one domain. In this work, we have proposed an algorithm for a fully-automatic ontology mapping that does not require any human intervention. To identify similar concepts, our algorithm is based on the combination of two parallel modules that are both based on linguistic analysis. The former analysis the concepts' names, while the later analysis their properties. This combination is very important to enhance the performances of the proposed algorithm. The linguistic analysis is it self based on the combination of a lexico-semantic similarity measure. At the end, the concepts considered as similar by combining the two previous results are mapped to each other through the relation "is-a". This provides a bridge ontology that handles the interoperability between the source ontologies that represent the same or complementary application domains.

Our algorithm is far from complete, several improvements must be completed to make it more efficient. In future work, we aim to enhance the mapping discovery stage by using other information retrieval techniques and elaborate and use a thesaurus of synonymy specific to the application domain, to enhance the results of the semantic similarity measures. Then, we will choose and study an appropriate application domain, on which we will apply our approach.

REFERENCES

[1] T. Gruber (1993). A translation approach to portable ontology specifications. Knowledge acquisition, vol. 5. pp. 199-220.

[2] S. Aubry (2007). Annotations et gestion de connaissances en environnement virtuel collaboratif. Thèse de doctorat. Université de technologies de Compiegne.

[3] A. Saleem, "Semantic Web Vision: survey of ontology mapping systems and evaluation of progress", Master Thesis, Intelligent Software Systems, Thesis no: MCS-2006:13, School of Engineering, Blekinge Institute of Technology, Box 520, SE – 372 25 Ronneby, Sweden, 2006.

[4] N.F. Noy (2004). Semantic integration: A survey of ontology-based approaches. SIGMOD Rec., Vol. 33, No. 4. pp. 65-70.

[5] M. Benerecetti, P. Bouquet and C. Ghdini, " Contextual reasoning distilled". Journal of Théoritical and Experimental Artificial intelligence, 12(3):279-305. 2000.

[6] A. Maedche, S. Staab (2002). Measuring similarity between ontologies. In proc of the European conference on knowledge acquisition and management-EKAW-2002, Madrid, Spain, October 1-4, LNCS/LNAI 2473, Springer, pp. 251-263.

[7] J. Euzenat and P. Shvaiko Ontology Matching, 2007, ISBN: 978-3 540-49611-3, Springer Berlin Heidelberg, New York

[8] V. I. Levenshtein, (1966). Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl., 6:707-710, 1966.

[9] M. A. Jaro (1989). « Advances in record linking methodology as applied to the 1985 census of Tampa Florida », dans Journal of the American Statistical Society, vol. 84, no 406, 1989, p. 414-420

[10] W. E Winkler (2006). « Overview of Record Linkage and Current Research Directions », dans Research Report Series, RRS, 2006.

[11] W. Hamming Richard (1950), "Error detecting and error correcting codes", Bell System Technical Journal 29 (2): 147–160, 1950.

[12] G. Stemme and A. Maedche, « Ontology Merging for Federated Ontologies on the Semantic Web". In

[13] proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001), Viterbo, Italy, 2001.

[14] N. F. Noy, M. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment". In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00), Austin, TX, USA, 2000.

[15] D. McGuiness, R. Fikes, J. Rice and S. Wilder. "An Environment for Merging and Testing Large Ontologies », In Proceedings of the 17th International Conference on Principales of Knowledge Representation and Reasoning (KR-2000), Colorado, USA, 2000.

[16] A. Doan, J. Madhaven, P. Domingos and A. Halevy, "Learning to map between ontologies on the semantic web". In proceedings of the 11th International World Wide Web Conference (WWW 2002), Hawaii, USA, 2002.

[17] P. Mitra and G. Wiederhold, « Resolving terminological heterogeneity in ontologies". ECAI'02 workshop on ontologies and semantic interoperability, Lyon, France, 2002.

[18] M. Fahad, N. Moalla,and A. Bouras, (2010). "Disjoint-Knowledge Analysis and Preservation in Ontology Merging process". Proceedings of 5th International Conference on Software Engineering Advances (ICSEA'10), IEEE Computer Society, August 22-27, Nice, France.

[19] M. Cho, H. Kim, P. Kim, "A new method for ontology merging based on concept using WordNet", Conference on Advanced Communication Technology, proceeding of ICACT' 2006, vol. 3, 2006.

[20] N. Choi, I. Y. Song, H. Han, "A Survey on Ontology Mapping", College of Information Science and Technology, Drexel University, Philadelphia, PA 19014.

[21] C. Ghidini and F. Giunchiglia, (2004). "A semantics for abstraction". In the proceedings of ECAI, pp.343.347.

[22] M. Ushold and M. Grüninger, « Ontologies and semantics for seamless connectivity », SIGMOD Record, 33(3), 2004.