Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 403 – 412

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Towards ontology matching based system through terminological, structural and semantic level

Aroua Essayeh[a,]* ,Mourad Abed[a]

[a]LAMIH, University of Valenciennes and Hainaut-Cambrésis, Le Mont Houy, 59313 Valenciennes cedex 9,France

Abstract

Ontology is a new paradigm introduced with the semantic web to describe in an explicit and formal way the various aspects of knowledge of a specific field. For this purpose, a single ontology may not be comprehensive to represent all due to the lack of a common and shared ontology between communities. Ontologies need to establish a number of interlinks to ensure communication between them, which is not always obvious because of their terminological, syntactic and semantic heterogeneity.

The proposed matching system aims to discover in an automatic way, the correspondence links between two intrinsically heterogeneous ontologies, through different techniques of calculations of similarity between their entities .It allows to reveal on one hand the issue of searching for the most relevant, coherent and meaningful alignments and on the other hand, to propose a new strategy that ensures flexibility and scalability of the system by the combination of the matchers.

Peer-review under responsibility of KES International

Keywords: Ontology, Ontology matching, ontology alignment, semantic heterogeneity, Semantic web.

* Corresponding author. Tel.: 0033648933798
E-mail address: Aroua.Essayeh@etu.univ-valenciennes.fr

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2015.08.154

1. Introduction

During the last decade, a lot of research has been undertaken in the field of the design and implementation of ontologies based matching systems. The concept of ontology has emerged in several fields of research allowing the establishment of correlation links between them; this promotes the use of matching systems.

Despite this diversity, the creation of a matching system is still a tedious task and challenges have not all been identified. As discussed in [1], the first challenge is related to the difficulty of evaluating the matching systems and their efficiency in addressing a wide automation process, the second challenge deals not only with the necessity of achieving a better implementation time of matchers by applying some strategies such as parallelization, distribution, approximation, modularization of ontologies and optimization of matching results, but also deals with the optimization of resources consumption such as memory consumption. Likewise, sometimes the matching process requires an external knowledge to extract the relations between the ontology entities, hence we must handle with this new information and ensure the combination of matchers in a complementary fashion and evaluated. However, in order to do that aforementioned challenge, it's obvious to give attention to the way of configuring the matcher both in the design and run time. Furthermore, even if the matching process is automatic, it is important to involve the user's decision to get comprehensive, reliable and accurate results of matching. Finally, in order to cope with these challenges, an interoperable infrastructure with a storage media and a reuse of alignments need to exist.

As mentioned earlier, these challenges can be divided into two main categories: First, those related to the alignments such as accuracy, consistency, expressiveness, versatility and significance. Second, ones related to the challenges of describing the matching systems' automation, the process optimization and the involved algorithms. These constraints are identified during the modelling phase (design time) or the phase of implementation process (run time).

In this article, we try to improve the research for alignments by applying algorithms, not only to extract all the possible alignments, but also to keep those who are most relevant.

The rest of the paper is organized as follows. the Section 2, fills in the concepts of matching, classifications, algorithms and the other approaches studied in the literature, then the third section will emphasize the more technical aspects dedicated to the developed tool, and finally the fourth section summarizes the conclusions, and points out the further researches.

2. Matching and Classifications

To overcome the problem of heterogeneous data from different sources, the matching process can provide a solution to this issue. Although, there are various ways to achieve this matching, they can be classified into 3 categories: either through the alignment when creating correspondences' links between the ontologies from different or similar fields, either by integration with the creation of a new ontology to be added to the two other ontologies belonging to different fields, or by the merger of ontologies where the domains are similar. In this article, we will focus on the first mode.

To emphasize the difference between the methods used in these modes, a state of the art is presented in [2] and enriched in this paper based on various dimensions. These dimensions describe the input and output data, the parameters and the resources [3], along which classifications are elaborated. The proposed approach falls within the classification mentioned in 1. However in the literature, several works have classified the matching algorithms into various levels.

In this context, the work of [4] presents two levels of classification (combined and individual approaches), another work of [5] classifies these approaches into two other levels; the first level concerns the rule-based approach and the second level tackles the learning-based approach. Furthermore, three types of classification are proposed by [[6], [3], [2]]; the first enhances the work of [4] by adding the semantic level. In 2007, it presents three new layers, the first is related to the context, the second describes the ontologies and the third deals with the data resources. All the levels mentioned above are linked to the layer that defines the field of knowledge. In 2013, he integrates the formal and informal methods to highlight the context based use of ontologies. [7] gathered the techniques of matching into 3 levels which are the syntactic, pragmatic and conceptual level to establishing the communication between the agents based systems.

Most of the above discussed classifications include three essential levels: terminological, syntactic and semantic level. Sometimes the definition of a concept is inherently vague and imprecise so it cannot be solved by classical matching methods. In this case, it is necessary to consider this imprecision by using fuzzy or probability methods.

3. Approaches and Matching Tools

The performance of matching process is widely used in several research projects. It is applied and implemented in different fields and ways of ensuring various tasks. Among the existing works, we can mention: [8] aims to address the needs of an educational resources sharing between multiple warehouses based on the concept of agents communicating with each other in order to perform the matching steps.

[9] uses the matching process to associate one element of model context with another from the ontology's field in order to annotate the task's model. This model is used then to enrich the BPMN for modeling the functional tasks of application. To reassemble and integrate the data distributed geographically, the works of [10] provides a mapping between a global ontology (global view of the system described with a common vocabulary) and local ontologies (defines each data source), however, four similarities measures are implemented (similarity of names, profile, structural and semantic) to perform this matching. In the field of intelligent network, [11] proposes a methodology for matching based on two levels; the similarity calculation level carried out by the elementary and structural based matching and the extraction alignments level.

We can assert that the essential role of matching is usually the sharing and integration of new data and promoting the interoperability.

In order to build our own system, it was overriding for us to study tools and platforms based on matching algorithms. We enriched the work of [1] which presented a state of art in matching tools.

It has been observed that the most common language of ontologies used with these systems is the OWL which is required by OAEI and also there are other languages such as RDF† and Skos‡ which are generated by the DSSim [12]and Falcon's tools [13]. On the other hand, other tools such as DSsim, Falcon, COMA ++ [14], RiMOM [15] can generate large-scale of ontologies by dividing such ontologies into blocks or parts (such as DSsim or Falcon) or by using heuristic rules for partitioning the input of schemes ontologies (COMA ++) or even by using different methods of similarity calculating at different levels (RiMOM, SystemBasedMatching [11], YAM ++ [16]). Furthermore, most of these tools focus on discovering 1:1 alignments like Sambo [17], Falcon, DSsim or RiMOM and others provide an n: m alignments with ASMOV [18] and AggrementMaker 's tool [19].

These systems can offer interfaces for the user in order to manipulate the matching, they can be a simple plugin integrated with other environments such as "protégé" or "WSMO" [20] tools with the plugin of "PROMPT" [21], "SAMBO" (integrated with KitAMO) or even a complete environment as COMA ++, DSsim, AgreementMaker.

4. Proposed System SIMTSS

The aim of our approach is to fulfil a dynamic research process for the alignment between the ontologies written in different languages and structuring including heterogeneous information. The result is new data stored as an XML file used in inference phases (query answering or integrating data).

The global architecture of the system baptized SIMTSS is composed of 5 layers bonded to a layer configuration showing the different APIs and tools used during the design and implementation of the system. The following figure shows the overall architecture of the proposed system:

† http://www.w3.org/RDF/
‡ http://www.w3.org/2004/02/skos/

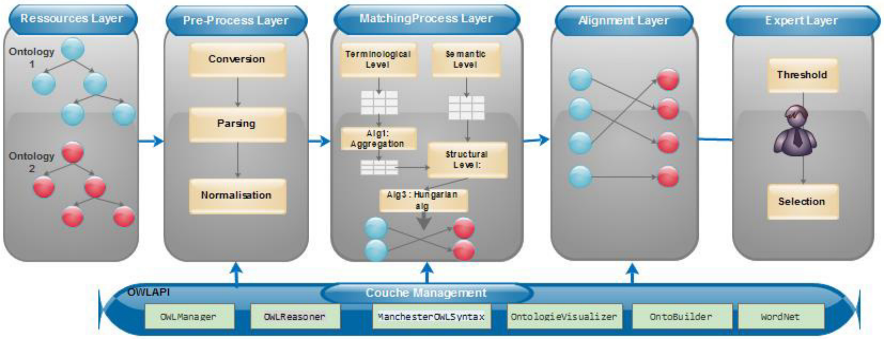

Fig. 1. Global architecture of the matching system SIMTSS

In the next part, we discuss every layer to explain how the system works.

4.1 The Resources Layer

This layer contains the source and target ontologies making the input of our system. They may be similar or of different types, written in RDF, SKOS, Turtle or other language. In this paper, we are not restricting ourselves to any particular language. The system integrates all the ontologies or partially in the matching process by mapping only the entities concerned by the matching (concepts, instances, properties...).

4.2 The Pre-Process Layer

This layer allows the standardization of ontologies in order to facilitate the task of matching. In this approach, [22] proposed to enrich the structure of ontologies written in XML by adding new elements since this format is only focused on the syntax and not on the semantics of the data. The main drawback of this method is that it is limited only to the tag's format; therefore, it can lead to new unnecessary information.

In our approach, we take as input two ontologies written in different languages. We consider in this step two parameters: the URL of the ontology or its physical location and the conversion's format. The format chosen by default is the OWL language for its capacity to make the ontologies more flexible, sharable and easily manageable.

The conversion step is not straightforward since it must migrate from one structure to another structure different from the first one.

The next step is the standardisation; it allows removing the punctuation signs, such as the spaces, numbers and special characters of ontologies element. Once these elements are distinguished (classes, properties, axioms, instances), the algorithm of normalization is applied and executed following four main methods: Lemmatisation, lowercase conversion, Stop_Words and Delete_Links. The result of this algorithm promotes the creation of alignments with more precision and less inconsistency.

4.3 The Matching Process

Aligning ontologies aims to find firstly the relationship between their entities, or rather to estimate the degree of similarity by calculating the similarity measures. In this article, we focused on establishing correspondence between ontology elements (classes, properties, instances, axioms) through the three main levels: terminological, structural and semantic. In our case, the properties are presented by the ObjectProperties and the DataProperties. The first type describes the relationship between the classes and the second type deals with the attributes associated with each concept or class.

In this regard, we define the process of matching as 4-uplet [23]:

$$M: < e1, e2, r, k > \quad (1)$$

With: e1, e2: represents the entities of the ontologie1 and 2, r: represents the semantic relationship between e1 and e2 such as (≡, ⊒, ⊑, ⊥) and k → [0, 1] represents the degree of resemblance between e1 and e2.

From this standpoint, we propose a new search correspondence which runs in hybrid and composite way [4] . Two ontologies are applied to test the different methods executed in every levels. The first one is “OntologyBook” and the second one is “OntologyProduct”, these ontologies are extracted from [2] and re-implemented with the protégé beta tool. We will discuss in the next section the different levels of matching.

4.3.1 The Terminological Matching

We propose a new methodology to search alignments between two entities belonging respectively to ontologies O1 and O2.

First, we focus to compare the labels related to concepts and Dataproperties. Then, we check the "DataTypes" based on the hierarchy developed by W3C[24], "domains" and "cardinality" of every Dataproperties. The aim is to assert that even if two properties have a high similarity as we have with the words "Year" and "Near", it is imperative to check their natures to avoid any kind of confusion. For terminological matching, five methods are used and combined to extract the most relevant alignments. Their measurements results are grouped into two classes:

1. If two entities are similar then the similarity's value belongs to the interval [0 1]. In this category, we apply the following methods:

JaroWinklerDistance

It measures the similarity between two strings. The longer the distance between them, the higher they are similar. It is calculated by the following formula:

$$d_w = d_j + (\delta_p(1 - d_j)) \quad (2)$$

With : $d_j = \frac{1}{3}(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m})$, δ_p : Prefix’s length, dj: Jaro’s distance, m: the number of corresponding characters, $|S_i|$: Length of the string, t: number of transpositions.

Table 1. JaroWinklerDistance’s *result*

	Author	Provider	Product	Creator
writer	0.555	0.638	0.436	0.662
publisher	0.611	0.569	0.417	0.417

The alignment established between the words “creator” and “writer” has the higher measure of similarity.

OverlapCoefficient.

This measure is related to the index of Jaccard measure calculated by defining the number of intersection divided by the smaller length of two concepts:

$$\text{overlap}(e_1, e_2) = \frac{|e_1 \cap e_2|}{\min(|e_1|, |e_2|)} \quad (3)$$

Table 2. OverlapCoefficient’s *result*

	Author	Provider	Product	Creator
Writer	0.0	0.0	0.0	0.0
publisher	0.0	0.0	0.0	0.0

For this example, overlapCoefficient don't provide any possible alignment between entities.

LevenshteinDistance.

This function calculates the minimum number of edit operations allowing the transformation of the entity e1 in entity e2. The distance is obtained by dividing this number by the minimum size of the two entities.

Table 3. LevenshteinDistance's *result*

	Author	Provider	Product	Creator
Writer	0.0	0.5	0.857	0.571
publisher	0.666	0.666	1.0	0.888

The alignment between «publisher» and «product» has the highest similarity value. In this case, they are considered the least similar while "writer" and "author" has the minimum value, so they considered as the most similar concepts.

TriGram.

The N-gram method is generally developed in the field of linguistic computing and probability. It is a contiguous sequence of n items from a given sequence of text or speech. In our system, we take n = 3, hence:

$$\mathrm{TG}(\mathrm{e1},\mathrm{e2}) = \frac{|\mathrm{trigram(e1)} \cap \mathrm{trigram(e2)}|}{\min(|\mathrm{e1}|,|\mathrm{e2}|)-2} \tag{4}$$

Table 4. Trigram's *result*

	Author	Provider	Product	Creator
writer	0.066	0.125	0.0	0.062
publisher	0.055	0.166	0.052	0.052

With Trigram's method, the concepts "author" and "writer" are the most similar.

2. If two entities are similar then the comparison value of their strings is equal to 1 if not 0. The method applied in this class is: JaccardDistance

JaccardDistance.

This method calculates the ratio between the intersection and the union's cardinal.

$$\mathrm{J}(\mathrm{e}_1,\mathrm{e}_2) = \frac{|\mathrm{e}_1 \cap \mathrm{e}_2|}{|\mathrm{e}_1 \cup \mathrm{e}_2|} \tag{5}$$

The Jaccard distance measures the similarity between the sets. It simply consists of subtracting the Jaccard indexed to 1.

$$\mathrm{J}_\delta\ (\mathrm{e}_1,\mathrm{e}_2) = 1 - \mathrm{J}(\mathrm{e}_1,\mathrm{e}_2) = \frac{|\mathrm{e}_1 \cup \mathrm{e}_2| - |\mathrm{e}_1 \cap \mathrm{e}_2|}{|\mathrm{e}_1 \cup \mathrm{e}_2|} \tag{6}$$

Table 5. JaccardDistance's *result*

	Author	Provider	Product	Creator
Writer	0.0	0.0	0.0	0.0
publisher	0.0	0.0	0.0	0.0

The result of this method doesn't show any similarity measure.

Terminological Aggregation.

The similarity values of each method are structured in a matrix.

For i from 1 to 5, the local matrix called $M_{local\ (i)}$ is determined for each terminological method. These matrices are combined to obtain a global matrix named M_{global}.

[2] discuss several aggregation‘s methods like Weighting, Voting or Arguing.

In our approach, weighting sum’s method is used to weight the five terminological methods; this weight is calculated referring to their execution time. To be more precise, we run this algorithm 5 times, and we took then the average of their results. However, the weight associated to each method is determined by dividing the execution’s time of each method by the total time:

$$w_i = \frac{T_i}{T_{Total}} \text{ , with } \sum_{i=1}^{5} w_i = 1 \tag{7}$$

The global matrix M_{global} is calculated using the following aggregation method:

$$M_{global} = \sum_{i=1}^{5} w_i \,.\, M_{local(i)} \tag{8}$$

After the execution of our algorithm we obtain the following table. It shows the results of weighting for each method:

Table 6. Weighting method

Methods	Execution time (ms)	weight wi (i=1..5)
JaccardDistance	6.8	0.395
JaroWinkler Distance	3	0,174
LevenshteinDistance	0,3	0 ,01
DiceCoefficient	1.4	0.082
TriGram	6	0.348

We obtain the result of the aggregation described in this table:

Table 7. Aggregation’s result

	Author	Provider	Product	Creator
writer	0.119	0.159	0.075	0.137
publisher	0.125	0.161	0.091	0.091

After applying the aggregation method, we note that the concepts “Provider" and "Publisher", "creator" and "writer" and "publisher" and "writer" are the most similar concepts. This result is very satisfactory for us and it shows the efficiency of all used methods and especially the aggregation one.

*4.3.2 **The Semantic Matching**.*

The semantic matching is used to complement the terminological methods matching since they were insufficient, they can compare only textual entities while neglecting their semantic and designations. Thereby, we use the WordNet as an electronic lexical database written in English. This resource is integrated in our system to find all the synonyms, hypernyms and hyponyms for each concept.

For this attempt, we identify two types of semantic matching:

1. Semantic Matching between concepts noted "MSC"
2. Semantic Matching between properties noted "MSP"

For each case, we establish our assumptions, for the first one:

- If two concepts defined by WordNet have synonyms in common then these two concepts are similar. We calculate the degree of similarity as follows, e_S and e_T are two terms with $\sigma: S \times T \rightarrow [0\ 1]$

$\sum$: The set of synonyms for every concept, the similarity is calculated as follows:

$$\sigma(e_S, e_T) = \frac{|\sum(s) \cap \sum(t)|}{\min(|\sum(s), \sum(t)|)} \tag{9}$$

- For other cases, they are not similar

The next table shows the result of this method

Table 8. MSP's result

	Author	Provider	Product	Creator
writer	1.0	0.0	0.0	0.0
publisher	0.0	0.0	0.0	0.0

Only the concepts "writer" and "author" have synonyms in common and their similarity value is equal to 1.

For the second type of matching, we find in the classification of [25] three types of calculation's methods of similarity based on WordNet (Edge-based methods, information-based Statistics Methods, Hybrid Methods). In this article, we propose applying the hybrid method. This method combines both Path's method, Wup's method and Lin's method.

The chosen methods are selected among several ones discussed in [25] to ensure two principals roles: The first one calculates distance between two words and their positions in the taxonomy with the methods Wup and Path. The second role calculates the probability of the word's appearance in the taxonomy based on information theory with Lin's method. The aggregate formula is:

$$\text{sim}\ (e1, e2) = \frac{\sum(\text{sim}_{\text{Path}} + \text{sim}_{\text{Wup}} + \text{sim}_{\text{Lin}})}{3} \tag{10}$$

Table 9. Semantic matching's result

	Author	Provider	Product	Creator
writer	1.0	0.390	0.287	0.085
publisher	0.086	0.082	0.096	0.085
Book	0.263	0.234	0.369	0.079
translator	0.333	0.309	0.222	0.085

This method allowed us to clearly identify the best alignment; the concept "writer" and "author" are the most similar from all the concepts. Also the concepts "translator" and "author" have a higher measure of similarity.

4.3.3 The Structural Matching.

The results obtained by the terminological and semantic methods are used as inputs for the initial structural mapping method. We analyse at this level, the internal structure of ontologies, they can be considered as graphs or taxonomies to specify the relationship between ontology concepts [11]. But this technique has already been treated in the semantic matching. For this purpose, we are interested in algorithms based on graphs. We adopt the SimilarityFlooding algorithm (SF) [26] that allows examining the alignments between the nodes of the graphs and their neighbourhood based on the notion of fixed point computation.

The improvement added to this algorithm is that the initial mapping of the SF is the result of terminological and semantic levels. It runs following five steps:

1 Transforming two ontologies in two graphs G1 and G2 with G1= (V1, E1), G2= (V2, E2) and V1: vertex, E1: Node 1 of the first graph and V2: vertex, E2: Node 2 of the second graph. All ontology's elements such

as OWLClass, DisjointClass, Property, OWLAnnotationProperty, OWLOntologyProperty...) become nodes except OWLObjectProperty that transforms into links connecting these nodes.

2 InitInitial Map: Take as input the global matrix obtained by the calculation of terminological and semantic measures.

3 Calculate the measure of initial propagation σ^0 (e1, e2) = 1.0 - Sim (e1, e2)

4 Continuing calculation of this measure to reach all nodes of the graph

5 Calculate the final matrix containing the similarity measures obtained by SF.

The next table shows the result

Table 10. Matching structural result

	author	Age	nom	Provider	provides
OntologyBook:writer	0.09	0.0	0.0	0.0	0.09
OntolgyBook:hasWritten	0.0	0.047	0.145	0.145	0.0

This table shows an extract of the result obtained by the structural matching; this result will be complement with the phase of alignment extraction.

4.4 Extracting Alignments

The SF algorithm presents the calculations of measures in the matrix format; we look through these measures to find the best alignments. Different methods can be applied to matrices as Karp algorithm or Minimum Cost Flow (MCF) or as Hungarian algorithm. The last one is an algorithm which finds an optimal assignment for a given cost matrix [27]. Our contribution aims to apply the Hungarian algorithm with ontologies to highlight the most correct matchers and to eliminate less relevant ones. This algorithm runs in five steps.

The obtained alignments are stored as an .xml file containing the two entities matching, similarity relationship and similarity's value between them. This structure is provided by the tool AlignmentAPI and integrated in our system to be used later in the inference phase.

4.5 Expert and Configuration Layer

In Expert level, the expert who is a knower of the field can confirm, suggest other alignments or ignore the results already obtained. The configuration Layer contains all the tools applied in the system as owl API, the reasoning OWLReasoner, ManchesterOWLSyntax, OntologieVisualizer, ontoBuilder and the external resource WordNet.

5 Conclusion

In this article, we present our approach for performing the ontology matching. The contribution of this paper is observed in the hybridization of several types of matching to keep the most relevant alignment. For this purpose, different algorithms have been adopted to meet the specific structures of ontologies in an elementary and structural way. The first manner is managed in semantic and terminological level and the second one is performed by using the concept of graph.

The similarity measures given by the two first levels are combined into a global matrix using a weighting sum aggregation and based on the criterion of the execution times of the methods. This step optimizes the performance of the implemented system. The WordNet dictionary provides not only the ontological entities synonyms but also helps to infer their degree of similarity based on hypotheses mentioned above. Furthermore, we adopt the SimilarityFlooding algorithm to analyze the internal ontology structures and to take into account the result obtained by global matrix as input of this algorithm.

To extract the best and relevant alignment from all these measures, we use the Hungarian algorithm. This

algorithm is performed with ontologies and ameliorated in this paper responding to the process of matching.
The future work aims to improve the task of the expert to make decision and to increase the number of ontologies in the process of matching. Finally, the result of this approach seems satisfactory but we need compare it with others in the same context to evaluate its relevance and accuracy.

References

1. Pavel, S.; Euzenat, J. Ontology matching: state of the art and future. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, ISSUE: 1, JANUARY* **2013,** pp.158-pp.176.
2. Euzenat, J.; Shvaiko, P. *Ontology Matching;* Springer Heidelberg New York Dordrecht London, 2013; pp 73,74,75.
3. Shvaiko, P.; Euzenat, J. *Ontology Matching-First Edition;* Springer Heidelberg New York Dordrecht London, 2007.
4. Rahm, E.; Bernstein, P. A. A survey of approaches to automatic schema matching. *The VLDB Journal 10,Digital Object Identifier (DOI) 10.1007/s007780100057* **2001,** 334–350.
5. Doan, A.; Halevy, A. Y. Semantic-Integration Research in the Database Community A Brief Survey. *AI Magazine Volume 26 Number 1 :Special issue on Semantic integration (© AAAI)* **2005,** 83-201.
6. Shvaiko, P.; Euzenat, J. *A Survey of Schema-based Matching Approaches;*, 2005.
7. Zanobini, S. *Semantic coordination: the model and an application to schema matching;* PhD thesis,University of Trento: Trento, Italy, 2006.
8. Elbyed, A. Un algorithme de mapping d'ontologies pour l'intéropérabilité d'entrepots de ressources pédagogiques. *1ères Rencontres Jeunes Chercheurs en EIAH, RJC-EIAH'2006*, 2006.
9. Marçal de Oliveira, K.; Bacha, F.; Mnasser, H.; Abed, M. Transportation ontology definition and application for the content personalization of user interfaces. *Expert Systems with Applications , www.elsevier.com/locate/eswa* **2013,** 3145-3159.
10. Shi, L.; Setchi, R. Ontology-based personalised retrieval in support of reminiscence. *Knowledge based system 45* **2013,** 47-61.
11. Santodomingo, R.; Rohjans, S.; Uslar, M.; Rodríguez-Mondéjar, J. A.; Sanz-Bobi, M. A. Ontology matching system for future energy smart grids. *Engineering Applications of Artificial Intelligence* **2014,** 242-257.
12. Nagy, M.; Vera, M. V.; Stolarski, P. DSSim Results for OAEI 2009. **2009,** 160-169.
13. Jian, N.; Hu, W.; Cheng, G.; Qu, Y. FalconAO:Aligning Ontologies with Falcon. **2005**.
14. Massmann, S.; Raunich, S.; Aumuller, D.; Arnold, P.; Rahm, E. Evolution of the COMA Match System. **2011**.
15. Zheng, Q.; Shao, C.; Li, J.; Wang, Z.; Hu, L. RiMOM2013 Results for OAEI 2013. **2013**.
16. Hoa Ngo, D.; Bellahsene, Z. YAM++ : (not) Yet Another Matcher for Ontology Matching Task. *BDA'2012 : 28e journées Bases de Données Avancées, Oct 2012, Clermont-Ferrand, France. pp.N/A, 2012*, 2012.
17. Lambrix, P.; Tan, H. SAMBO—A system for aligning and merging biomedical ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web 4* **2006,** 196-206.
18. Jean-Mary, Y. R.; Kabuka, M. R. ASMOV: Ontology Alignment with Semantic Validation., 2006.
19. Cruz, I. F.; Palandri Antonelli, F.; Stroe, C. Using AgreementMaker to Align Ontologies For OAEI 2011. **2010**.
20. Dimitrov, M.; Simov, A. *WSMO Studio Users Guide;*, 2006.
21. Fridman Noy, ; Mark A, M. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment;*, 1997.
22. Cruz, C.; Nicolle, C. RDF rules for XML Data conversion to OWL Ontology. **2009**.
23. Giuseppe , P.; Talia, D. UFOme: An ontology mapping system with strategy prediction capabilities. *Data & Knowledge Engineering* **2010,** *Volume 69* (Issue 5,), 444-471.
24. Paul V, B.; Malhotra, A. *XML Schema part2: Datatypes Second Edition.W3C Recommendation 28 Octobre 2004. World Wide Web Consortium(W3C);*, 2004.
25. Feiyu, L.; Kurt, S. A Survey of Exploiting WordNet in Ontology Matching. **2008**.
26. Melnik, S.; Garcia-Molina, H.; Rahm, E. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. *In Proceedings of the 18th International Conference on Data Engineering (ICDE '02). IEEE Computer Society, Washington, DC, USA, 117-.* **2002**.
27. Munkres., K. &. *Bipartite Matching & the Hungarian Method;*, 2006.