Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 23 (2011) 558 – 564

Procedia Engineering

www.elsevier.com/locate/procedia

2011 International Conference on Power Electronics and Engineering Application (PEEA2011)

A new bidirectional method for ontologies matching

Hacene Belhadef[a] a*

[a] Department of Computer Sciences, Mentouri University of Constantine, Road of Ain ElBey 25017, Algeria

Abstract

Recently, much research has focused on developing techniques for schema/ontology matching and mapping as it is required in many areas e.g. heterogeneous database integration, merging of ontologies, semantic query processing. In this paper, we present a novel method to detect and repair the list of homologs concepts; this last is based on the bidirectional comparison (Checking) between the two matched ontologies to filter this list of concepts. The used measures are classical just to show the reliability and realisability of our method, but we can generalize it for supporting other advanced measures.

Keywords: bidirectional verification; ontologies; ontology matching; similarity measures

1. Introduction

From integration of database schemas to the alignment (matching) of ontologies, the problem that generated the most difficult issues to resolve in recent years, that is the search for correspondence (between database schemas, schemas or XML documents or between ontologies). Different solutions exist to identify common concepts and involve different sources using an ontology. Three approaches can be taken [1]: the overall approach, the multiple approaches and hybrid approach [2].

The *schema matching* process identifies correspondences between elements from different schemas. The *schema mapping* process defines these correspondences i.e. provides view definitions that link the two

* Corresponding author. Tel.: +213-31-818494; fax: +213-31-818494.
E-mail address: hacene_belhadef@umc.edu.dz

doi:10.1016/j.proeng.2011.11.2547

schemas [3]. Schema matching and mapping may generally be undertaken manually, semi-automatically or automatically. Manual or semi-automatic schema matching and mapping is tedious, complex and time consuming task because of open and heterogeneous nature of data. The volumes and heterogeneity of data, in particular, mandate the development of automatic schema matching and mapping techniques. Automatic schema matching and integration is required to reduce manual efforts but require more research and development for high quality techniques and results [4].

Nowadays, the interested practitioner in ontology mapping, is often faced with a knotty problem: there is an enormous amount of diverse work originating from different communities who claim some sort of relevance to ontology mapping. For example, terms and works encountered in the literature which claimed to be relevant include: *alignment, merging, articulation, fusion, integration, morphism*, and so on. Given this diversity, it is difficult to identify the problem areas and comprehend solutions provided. Part of the problem is the lack of a comprehensive survey, a standard terminology, hidden assumptions or undisclosed technical details, and the dearth of evaluation metrics [5].

An ontology is considered here as a taxonomy of concepts and the problem of matching is reduced to: “for each concept node in one taxonomy, find the most similar node in the other taxonomy”. Our concern discussed in this paper is the reliability of the results and how to make corrections to eliminate the matching concepts when the degree is low.

2. Related work

In the research literature, there are many diverse approaches, techniques and systems for alignment, mapping and merging of heterogeneous ontologies.

Various works have been developed for supporting the mapping of ontologies. An interesting survey which gathered 35 works is presented in [5]. In [6] we can find other surveys on ontology alignment. In most approaches heuristics are described for identifying corresponding concepts in different ontologies, e.g. comparing the names or the natural language definition of two concepts, and checking the closeness of two concepts in the concept hierarchy. We can also find more researchers of schema/data matching and mapping area that have done considerable efforts [3][7][8][9][10][11][12][13][14].

3. Ontology

The term "ontology" comes from the field of philosophy that is concerned with the study of being or existence. In philosophy, one can talk about ontology as a theory of the nature of existence. In the context of computer and information sciences, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members) [15].

The preceding definition leads to a set of definitions that can be used as a basis for algebraic formulation of the term Ontology and its components [16]:

Definition 1. **A term** is a triple $\tau = [\eta, \delta, A]$, $\tau \in T$, where η is a string of characters containing the name of the term, δ is a string of characters containing its definition and A is a set of attribute domains $A_1, A_2, ..., A_n$, each associated to a value set V_i.

Definition 2. **A relation ϕ** : $T \rightarrow T$, $\phi \in \Phi$, : is a function from T to T such that for every term $\tau_1 \in T$, there is a term $\tau_1 = \phi(\tau_1), \tau_2 \in T$.

Definition 3. **A semantic relation σ** between two terms is a relation that belongs to the set of semantic relations $\Sigma = \{$Hypernymy, Hyponymy (is-a), Mereonomy (part-of), Synonymy $\}$, $\Sigma \subset \Phi$.

Definition 4. **A spatial relation** ρ between two terms is a relation that belongs to the set of spatial relations P= {adjacency, spatial containment, proximity, connectedness}, $P \subset \Phi$.

Definition 5. An ontology is a pair $\Theta=[T, \Phi]$, where $T=\{\tau_1, \tau_1, ..., \tau_n\}$ is a set of terms, and $\Phi=\{\Phi_1, \Phi_2, ..., \Phi_n\}$, and $\exists \phi_i \in (\Sigma \cup K)$.

4. Matching and Mapping

The *schema matching* process identifies correspondences between elements from different schemas. The *schema mapping* process defines these correspondences i.e. provides view definitions that link the two schemas [3]. Schema matching and mapping may generally be undertaken manually, semi-automatically or automatically. Manual or semi-automatic schema matching and mapping is tedious, complex and time consuming task because of open and heterogeneous nature of data. The volumes and heterogeneity of data, in particular, mandate the development of automatic schema matching and mapping techniques. Automatic schema matching and integration is required to reduce manual efforts but require more research and development for high quality techniques and results.

Ontology alignment, or ontology matching, is the process of determining correspondences between concepts. A set of correspondences is also called an alignment. *Ontology matching* is a promising solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities of the ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or for navigation on the semantic Web. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate [6].

In a formal way, alignment is defined by the map function as follows:

$$map : O_1 \rightarrow O_2 \quad such\ as\ \ map(e_1)=e_2\ \ si\ sim(e_1,e_2)>=t \qquad [1]$$

Where O_1 and O_2 are the two ontologies to align, t is a minimum threshold of similarity belonging to the interval [0, 1], $e_1 \in O_1$ and $e_2 \in O_2$. e_1 and e_2 represent the entities (concepts or relations) in the two ontologies. The threshold t indicates the minimum level for two entities are similar.

The alignment of two ontologies, O_1 and O_2 amounts to determining the correspondence between the various ontological entities by category (type). All methods of alignment determine the correspondences between ontological entities using measures of similarity [17]. The measures of similarity or distance (dissimilarity) allow evaluate the similarity or the distance between two elements (or individuals).

5. Our new method

Several works have been realized to provide matching algorithms, to solve the problem of the correspondences or incompatibilities of schemas. Our approach is based and characterized by a bidirectional verification (checking) to reduce the number of matches having the weak and ambiguous, so to increase the quality of the results. As we have already said, that our approach is one that takes into account the textual description of the entities, whether in terminological or structural level.

The proposed algorithm is divided into four stages (see Fig. 1). The first is the normalization (preprocessing), the second is the syntactic verification, the third is the structural verification and the fourth is bidirectional verification, this last is used in complement of the second and third step. The scheduling of this algorithm is not sequential and the order of execution of steps 2 and 3 is to the choice of the user who decides what stage will begin. The notations used in the developed algorithm are summarized in Table 1.

Table 1. Notations used in the algorithm

The symbols used in the algorithm are :

- O_1, O_2 : The two ontologies to align in OWL format
- n represents the number of concepts of the ontology O_1
- m represents the number of concepts of the ontology O_2
- N_{1i} : the node of the ontology O_1 $\{\forall\ i \in [1..n] / N_{1i} \in O_1\}$
- N_{2j} : the node of the ontology O_2 $\{\forall\ j \in [1..m] / N_{2j} \in O_2\}$
- SimSyn : syntactic similarity
- SimStr : structural similarity
- MSsyn : Syntactic similarity matrix of size (n, m)
- MSstr : Structural similarity matrix of size (n, m)

The proposed alignment algorithm (see Fig. 1 and Fig. 4), takes as input two ontologies in OWL format, source ontology (O_1) and a target ontology (O_2), and provides as output an XML file. The algorithm is based on two main stages (not successive). The first stage, implemented through the function *FuncSyn()* that allows calculates the syntactic similarity between the two ontologies, the second stage is implemented by the function *FuncSruct()*, to calculate the structural similarity. We can add a third stage to measure the semantics between the two aligned ontologies, but this problem was not addressed in this article. In this algorithm we consider only the alignments in which the relations are equivalence (≡) or subsumption (⊆,⊇) between the named entities in each ontology (simple alignment).

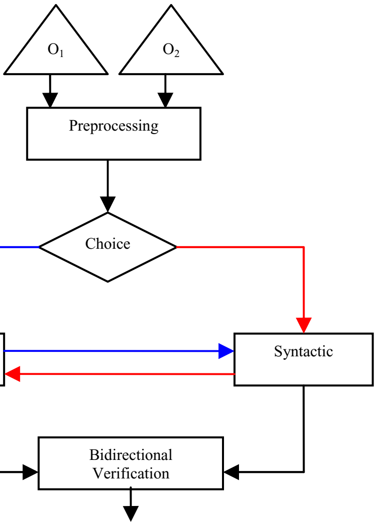
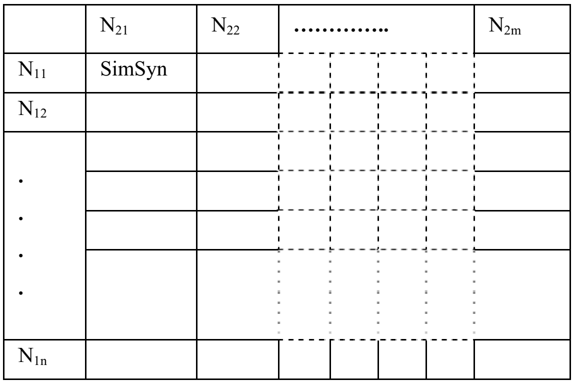


Fig. 1. Organigram of the algorithm

Stage 1: Normalization (Preprocessing): This step aims to achieve a preprocessing on schemas of ontology. It takes as input an OWL ontology and analyze it through the parsers of ontologies schemas to simplify the contents of strings representing concepts in order to bring back them to equivalent formats [18]. In this step of terminological standardization, we perform several successive operations:

- The normalization of the case (upper / lower case) is to convert each letter of the string to lowercase (or uppercase).
- Remove spaces and replace them by the character "-" for each block of white.
- Replace the accents, cedilla, etc.., by standard characters of the alphabet and what to correct spelling errors by minimizing the problems of ambiguity.

Stage 2: Syntactical verification: The syntactical verification is done through the function *FuncSyn()*, which takes as input parameters the two ontologies O_1, O_2 and the threshold t1 set by the user. It provides at the end of its execution the SMsyn matrix (see Fig. 2) containing the different values of syntactic similarity calculated using the function *syntacticMeasure(N_1, N2)*. This function takes many forms and variations; it runs by using both a single similarity measure that is selected by the user via a graphical user interface (see Fig. 4).



	N_{21}	N_{22}	…………..	N_{2m}
N_{11}	SimSyn			
N_{12}				
. . . .				
N_{1n}				

Fig. 2. The similarities matrix

Table 2. (a) Algorithm of the syntactical function ; (b) Algorithm of the structural function

```
Function : FuncSyn().
 Inputs :
   1) $O_1$ and  $O_2$ : two ontologies to align
   2) $t_1$ : syntactic similarity threshold
 Output :
   SMsyn :syntactic similarity matrix
   begin
   For ($N_{1i}$ ∈ $O_1$) do /* travel the nodes of
                          the ontology $O_1$ */
   For ($N_{2j}$ ∈ $O_2$) do /* travel the nodes of
                          the ontology $O_2$ */
      SimSyn ← syntacticMeasure($N_{1i}$ ,$N_{2j}$)
      if SimSyn >= $t_1$   then
         SMsyn(i , j) ← SimSyn
   return (SMsyn)
  end
```

```
Function : FuncStuct().
 Inputs :
   1) $O_1$ and  $O_2$ : two ontologies to align
   2) $t_2$ : structural similarity threshold
 Output :
   SMstr: lexical similarity matrix
   begin
   For ($N_{1i}$ ∈ $O_1$ ) do /* travel the nodes of
                            the ontology O1 */
   For ($N_{2j}$ ∈ $O_2$) do /* travel the nodes of
                           the ontology O2 */
   SimStr ← StructuralMesure($N_{1i}$,$N_{2j}$)
   if  SimStr >= $t_2$   then
       SMstr(i , j) ← SimStr
   return(SMstr)
  end
```

Stage 3: Structural verification: In this step, we apply a structural similarity measure expressed by the function *StructuralMesure*(N_{1i},N_{2j}). This measurement is used through the function *FuncStuct()* that takes as input the two ontologies O_1, O_2 and the threshold t_2 that is always set by the user. It provides at the end of its execution, the matrix SMstr having a similar structure as that of matrix in Fig. 2 but with structural similarity values.

The function *StructuralMesure*(N_{1i},N_{2j}) compares the internal structure of the two concepts, N_1 and N_2 by checking the number, names of attributes and data types. The verification of the names that call to *syntacticMeasure*(N_{1i} ,N_{2j}), but this time takes as parameters two strings denoting the names of two attributes.

Stage 4: Bidirectional verification : This type of verification, is to make a matching in both directions of the two ontologies in question, the purpose of this step is to eliminate the ambiguity in the candidates found (homologs concepts). Fig. 3, shows that the concept C_{24} of target ontology (O_2) is a homolog of two concepts C_{14} and C_{15} of the source ontology (O_1). If we take into consideration the type of equivalence relations (≡), this means that the two concepts C_{14} and C_{15} are the same that is to say they represent the same physical entity, something that contradicts with the principle of ontology definition that does not accept the redundancy of concepts definition.

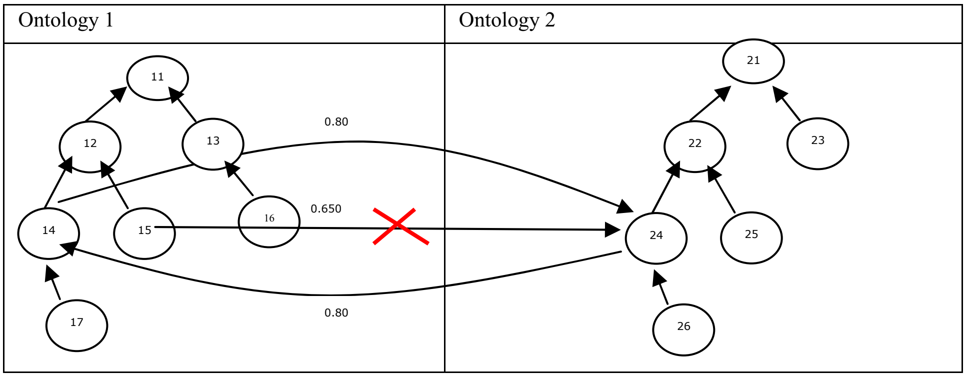


Fig. 3. Bidirectional verification

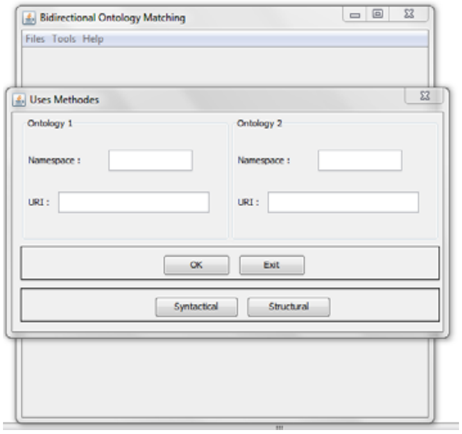


Fig. 4. The graphical user interface

6. Conclusion

In this paper we have presented a general description of our new method for matching ontology domain, this last is characterized compared with classical approaches by a bidirectional verification based on a terminological syntactical and structural verification, this verification allows to filter the list of homolog concepts and eliminate the ambiguity between ontologies. The proposed method can support the additional verification steps such as linguistic and semantic but the main goal here is not to describe a robust method of matching, but to show its feasibility in our system. So in the future work, we propose to generalize it by using the sophisticates similarity measures

References

[1] Sheeren D., Méthodologie d'évaluation de la cohérence inter-représentations pour l'intégration de bases de données spatiales. Thèse doctorat de l'université paris 6, 2005.

[2] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G, Neumann H and Hübner S. Ontology-based integration of information - A survey of existing approaches. *17th* International Joint Conference on Artificial Intelligence (IJCAI'01). 2001.

[3] Rahm E., Bernstein P. A. A Survey of approaches to Automatic Schema Matching, In: VLDB Journal- The International Journal on Very Large Data Bases, vol. 10, no. 4, pp. 334-350, ISSN: 1066-8888

[4] Review of ETL and Ontology/Schema Matching Techniques and Tool (2010), By: **Geo**scientific **D**ata **I**ntegration, At: geodi.ucc.ie/files/deliverables/GeoDI_D25_Final.pdf

[5] Kalfoglou Y., Schorlemmer M. Ontology Mapping: The State of the Art, The Knowledge Engineering Review Journal**,** vol. 18, no. 1, pp. 1-31**,** 2003.

[6] Euzenat J., Shvaiko P. Ontology Matching. Springer, Approx. 340 p., 67 illus., Hardcover, ISBN 978-3-540-49611-3 **,** 2007.

[7] Madhavan J., Bernstein P. A., Rahm E. Generic Schema Matching with Cupid. In: Proceedings of the 27th VLDB Conference, pp. 49-59, Roma, Italy. 2001.

[8] Aumueller D., Do H. H., Massmann S., Rahm E. Schema and Ontology Matching with COMA++, In: Proceedings of the 2005 ACM SIGMOD International Conference on Management Data, pp. 906-908, Maryland, USA. , 2005.

[9] Rizopoulos N. Automatic Discovery of Semantic Relationships between Schema Elements, In: International Conference on Enterprise Information Systems, pp. 3-8, Porto, Portugal. 2004.

[10] Shvaiko P., Euzenat J. A Survey of Schema-based Matching Approaches. Technical Report DIT-04-087, Informatica eTelecomunicazioni, University of Trento, Italy. 2005.

[11] Hakimpour F., Geppert A. Global Schema Generation Using Formal Ontologies. In: Proceedings of the 21st International Conference on Conceptual Modeling, pp. 307-321.2002.

[12] Embley D. W., Xu L., Ding Y. Automatic Direct and Indirect Schema Mapping: Experiences and Lessons Learned. In: ACM SIGMOD Record, pp.14-19. 2004.

[13] Zamboulis L., Poulovassilis A. Using AutoMed for XML Data Transformation and Integration. In: Proceedings of the 16th International Conference on Advanced Information Systems Engineering, third International Workshop on Data Integration over the Web,, vol. 3, pp.58-69. 2004.

[14] Naz T. Dorn J. Configurable Meta-Search in the Human Resource Domain- A Hybrid Approach to Schema and Data Integration for Meta-search Engines. Lambert Academic Publishing, ISBN:978-3-8383-0230-0. 2009.

[15] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications Knowledge Acquisition, 5(2):199-220. 1993.

[16] Fonseca F., Davis C., Câmara G. Bridging Ontologies and Conceptual Schemas Geographic Information Integration. Geoinformatica, Volume 7 Issue 4, 355-378. 2003.

[17] Zghal S., BenYahia S., Nguifo E. M., Slimani Y. SODA : Une approche structurelle pour l'alignement d'ontologies OWL-DL », 1ères Journées Francophones sur les ontologies, 18-20 Octobre 2007.

[18] Elbyed A. ROMIE : une approche d'alignement d'ontologies à base d'instances, Thèse doctorat de l'institut national des télécommunications de l'université Evry-Val d'Essonne, 2009.