World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:10, 2015

Resources-Based Ontology Matching to Access Learning Resources

A. Elbyed

Abstract**—**Nowadays, ontologies are used for achieving a common understanding within a user community and for sharing domain knowledge. However, the de-centralized nature of the web makes indeed inevitable that small communities will use their own ontologies to describe their data and to index their own resources. Certainly, accessing to resources from various ontologies created independently is an important challenge for answering end user queries. Ontology mapping is thus required for combining ontologies. However, mapping complete ontologies at run time is a computationally expensive task. This paper proposes a system in which mappings between concepts may be generated dynamically as the concepts are encountered during user queries. In this way, the interaction itself defines the context in which small and relevant portions of ontologies are mapped. We illustrate application of the proposed system in the context of Technology Enhanced Learning (TEL) where learners need to access to learning resources covering specific concepts.

Keywords**—**Resources query, ontologies, ontology mapping, similarity measures, semantic web, e-learning.

I. Introduction

NOWADAYS, it is important to provide to users the tools and systems allowing a transparent access to the huge of data and resources accessible via the Web. In this goal, ontologies are increasingly used. Ontologies are an important tool providing semantics to data and documents in several areas. They are used as basis for interoperability between systems and for data integration, by providing a common terminology over a domain.

In any considered area or domain, the number of existing ontologies is increasingly great, which are mostly complementary but could also contain important overlapping. In order to use the needed ontologies in an integrated way, "bridges", i.e. mappings, between ontologies must be built. Mapping two ontologies O_1 and O_2, means defining semantic relations between concepts of O_1 and concepts of O_2. Mappings are often established manually by the domain experts, but because the increase number of ontologies and their size, there is need to generate automatically (or semi-automatically) all mappings. Several algorithms and tools have been proposed, nevertheless due to the task complexity, none of them is completely satisfactory [1]-[3]. The problem is therefore still open.

To infer the links of similarity between two concepts or terms, several methods could be used. Almost existing ontology mapping algorithms use and combine several methods, mostly syntactic and linguistic methods. Some of them use information of the structure (topology) of considered ontologies. Very few use also information on documents or resources (instances) connected to the ontologies, and more precisely to concepts of the ontologies.

In this paper, we propose a system, called ROMIE (Resource based Ontology Mapping with an Interactive Environment), which is a system allowing, in one hand, users to make queries on distant resources via ontologies (ontologies which are linked up by mappings), and in another hand, to generate ontology mappings, using among other things, semantic relations between concepts which are inferred by the resources themselves. Thus, ROMIE takes on input a set of local resources and ontologies and considers also distant resources and ontologies which could be achieved via the Web. Each resource is connected to one or several concepts. When trying to answer queries on resources, ROMIE aims to improve user satisfactory by using local ontology information. Indeed, if there isn't any resource that suits the query or if the query result does not satisfy user goals, ROMIE try to find other resources belonging to distant ontology. To achieve this goal, ROMIE uses a mapping process in order to find for each concept of the query the corresponding one in the distant ontology in a transparent way. Thus, for each user query, the mapping process generates new mappings between concepts. After some number of iterations, all correspondences between the two ontologies are established.

The mapping process in ROMIE is based on the use of several methods of similarity measure between concepts including the classical syntactic, linguistic and structural methods. Nevertheless, an important aspect of ROMIE consists to use the resources in the mapping process. The resources are used to enrich the ontologies by semantic relations between concepts, and these relations are then used to improve the mapping process.

We consider here as application the context of Technology Enhanced Learning (TEL). The number of ontologies in TEL domain is growing considerably, as it is the case in several domains, such as biology for instance. An urgent need in TEL is the discovery of suitable resources and the marshalling of those resources to work together to perform a learning task. Ontology-driven TEL has concept-based representation of the specific subject domain and learning resources indexed by domain concepts. Communication between ontologies and interoperability are thus very important for TEL systems.

In the first part of the paper, we present the general principle of the mapping process. Then we focus on ontology

Abdeltif Elbyed is with the Laboratory of Computer Science and Decision Aids (LIAD). Department of Mathematics and Computer, Faculty of Sciences Ain-Chock, Hassan II University of Casablanca, BP5366, Maarif, Casablanca 20100, Morocco (e-mail: a.elbyed@fsac.ac.ma).

enrichment by semantic relations and their use during the mapping process. Finally, we describe the developed prototype and give some results of our query and mapping process obtained on the application of learning resources, before concluding.

II. Mapping Process General Principle

The mapping process is based on the measure of similarity between concepts of different ontologies. An important step of all existing algorithms and systems for mapping ontologies practice the syntactic and linguistic methods, in order to measure the similarity between concepts with a terminological point a view. Nevertheless, these methods are not sufficient for a good and appropriate measure. That's why other types of methods, namely structural (topological) methods, are often used. These methods take into account the structure of the ontology, mostly information on concept neighbourhood, i.e. fathers and children of the considered concepts (concepts to map). Besides these relations of "father" and "child" between concepts of the ontology, we consider, in our system, other kind of relations, namely semantic relations, which are very important in order to add a semantic level to the similarity. Unfortunately, existing ontologies have rarely this type of relations. Therefore, an important step of our system is to enrich the ontology by semantic relations between its concepts, before performing mappings. These relations are generated using information of learning resources connected by ontology concepts (Fig. 1). This step will be explained in the next section.

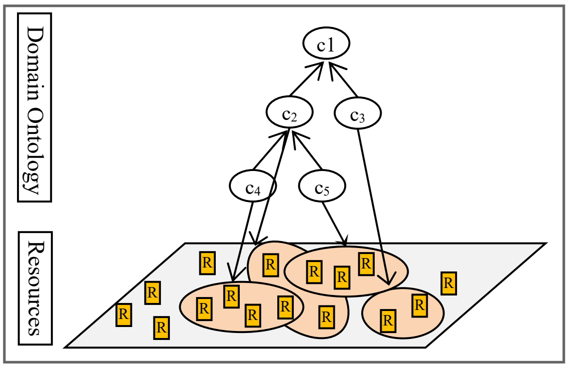


Fig. 1 Relation between learning resources and ontology concepts

The mapping process in our system considers as input two ontologies and a set of resources indexed by these ontologies. The mapping is then performed in several steps, each step develops one or several methods (syntactic, linguistic, structural and semantic):

- *Similarity values calculation*: syntactic and linguistic methods are applied on couples of concepts, in order to measure their terminological similarity. We use in our case several syntactic methods which calculate an edit distance between terms, and a linguistic method based on WordNet. Each of these methods returns a similarity value.
- *Generation mapping hypotheses*: the similarity values *returned* by the precedent step are then combined using a simple formula, in order to generate one similarity value *SV:*

$$SV = \frac{\sum_i Conf\ \eta i \times SV\ \eta i}{\sum_i Conf\ \eta i}$$

The **Confηi** and **SVηi** are respectively the confidence level of the method ηi and the similarity value returned by this method for a couple of concepts. Hypotheses of mapping are thus generated.

In this step, structural (topological) and semantic methods are also applied to improve the similarity values of mapping hypotheses and/or to generate new mapping hypotheses. They are based on existing relations between concepts and on mappings already established and validated.

Combining a lot of similarity methods can generate some wrong mapping hypotheses. We use filtering methods in order to eliminate false mapping hypotheses:

- *Mapping hypotheses filtering*: These methods are based on structural and semantic relations between concepts and on *mappings* already validated. A threshold is also used to eliminate mapping hypotheses with similarity values too low.

III. Ontology Enrichment Process

An important contribution of this paper is the ontology enrichment phase, which plays a crucial role to improve mapping results. As presented in second section, we use a hybrid approach of ontology mapping which mixes linguistic, structural and semantic approaches. The semantic approach exploits different semantic characteristics of ontologies. The problem is that the number and/or the quality of existing semantic relations in ontologies are in general very low. The main objective of our resource based approach is to exploit information about the resources connected to the ontology concepts. This information is analyzed to infer new semantic relations between ontology concepts. This process will enrich the semantic of the ontology.

The first step in this process is to analyze resources properties, in order to generate relations between them. The second step is to propagate theses relations from resources to concepts. To better understand these two steps, a brief introduction on the learning resource model developed in [4], [5]. It gives a semantic description of a resource. We consider that a resource is any digital object like a set of web pages, a file or a program (a simulator for example) in any format (e.g., text, video, image, audio, etc.). We just suppose that it is a unit accessible via an URI. Each resource is described by general metadata (e.g., author, title, language) and is indexed with the concepts of the domain ontology. In other words, it content develops one or more concepts. In addition, each resource may contain prerequisites (what is required by the resource) expressed by one or more concepts.

The following subsections present the two steps of enrichment.

A. Relations between Resources

Let us consider two resources *R* and *R'*. We note *Pre(R)* (respectively *Pre(R')*) the prerequisite of *R* (respectively *R'*) and *Cont(R)* (respectively *Cont(R')*) the content of R (respectively *R')*. The analysis of resources properties allows us to propose a set of semantic relations with characteristics. These relations (Table I) can be deduced automatically. The characteristics of the relations play allow the generation of new semantic links between learning resources as it is shown in the mapping process.

TABLE I
RELATIONS AND PROPERTIES BETWEEN RESOURCES

Relation name	Relation properties between resources R and R'	Relation characteristics
Strong-substitution	Pre(R) = Pre(R')	Symmetry and Transitive
Weak-substitution	Pre(R) ⊂ Pre(R')	Anti-symmetry and Transitive
Equivalence	R is substituable by R' &Cont(R)= Cont(R')	Symmetry and transitive
Weak-precedence	Cont(R) ⊆ Pre(R')	Anti-symmetry and transitive
Strong-precedence	Cont(R)=Pre(R')	Anti-symmetry and transitive
Strong-Crossed	Cont(R)=Pre(R') &Cont(R')= Pre(R)	Symmetry and transitive
Weak-Crossed	Cont(R)⊂ Pre(R') &Cont(R')⊂ Pre(R)	Anti-symmetry and transitive
Part of	Cont(R)⊂ Pre(R') &Cont(R)⊂ Pre(R')	Anti-symmetry and transitive
More general	Pre(R)⊃Pre(R') &Cont(R)=Cont(R')	Anti-symmetry and transitive
More specific	Pre(R)⊂ Pre(R') &Cont(R)=Cont(R')	Anti-symmetry and transitive
Mismatch	if there is no relation between their properties	Symmetry and Transitive

In Fig. 2, we present two examples of defined relations between resources, namely: substitution and weak-precedence relation.

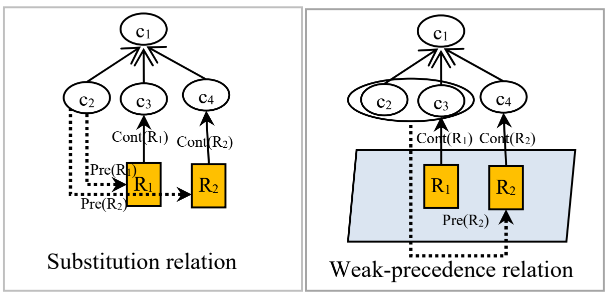


Fig. 2 Semantic relations examples between two learning resources

B. Generation of Semantic Relations Between Concepts

For each generated relation *rel* between two resources, an equivalent relation between associated concepts is generated (Fig. 3). However, this propagation depends on the number of resources associated to each concept. Thus we must consider the set of resources linked to each concept rather than individual ones. Each set of resources develops the same concept but with possibly different prerequisites. Therefore, we have to distinguish several subsets corresponding to equivalent resources (same content and same prerequisite). Given two concepts C_1 and C_2, each subset A_i of equivalent resources associated to C_1 may have a semantic relation rel_i with another subset B_i associated to the concept C_2. The problem is to decide which rel_i to propagate towards the corresponding concepts (to obtain a new relation between the concepts C_1 and C_2). Our proposition is to associate a weight to each rel_i depending on the similarity between the corresponding subsets A_i and B_i. In this purpose, we have used the well-known formula of Jaccard's similarity measure. The Jaccard measure is used to calculate the distance between two sets *A* and *B*; it takes the lowest value 0 when *A* and *B* are disjoint, and the highest value 1 when *A* and *B* are the same. Thus, we consider that the weight of a semantic relation between two concepts C_1 and C_2 is defined by Jaccard measure weight between the two sets of resources A_i and B_i related respectively to C_1 and C_2. In other words, the semantic relation between C_1 and C_2 is a tuple:$(C_1, C_2, \sigma_{reli}(Ai, Bi))$where $\sigma_{reli}(Ai, Bi)$is the Jaccard measure weight between the sets $A_i$and B_i and each element of A_i are related by rel_i relation to element of B_i.

Considering Я={$rel_1,\ldots rel_K$} as a set of semantic relations between two sets of resources A and B, we note $A \leftarrow rel_i \rightarrow B$. The Jaccard measure is calculated with the following formula:

$$\forall rel_i \in Я\, \sigma_{reli}\,(A,B)=\frac{\left|A \bigcap_{reli} B\right|}{\left|A \bigcup B\right|} \text{ with}$$

$$A \bigcap_{reli} B=\{a_j \cup b_k / \forall a_j \in A,\ \exists b_k \in B/\ a_j \leftarrow rel_i \rightarrow b_k\}= A_i \cup B_i$$

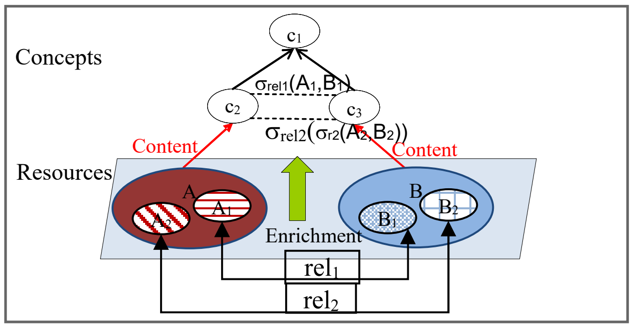


Fig. 3 Propagation of semantic relation between set of resources to semantic relation

The comparison of resources allows thus building more semantic relations between ontology concepts, which are then used in the mapping process as described below.

IV. RESOURCE-BASED MAPPING PROCESS

We study the mapping of ontologies through the ontology *morphism* properties. After introducing this *morphism* this section presents semantic matchers and filters through examples.

A. Ontology Morphism

The principle of ontology *morphism* is to consider that each ontology relation (hierarchical or semantic relation) between two concepts in the same ontology is equivalent to the relation between their images (i.e., mapping concept) in the other ontology. In other words, when two ontologies O and O' are mapped, a semantic relation between two concepts of O is equivalent to the relation between their images (by mappings) in O'. For example, if the concept c_1 precedes semantically the concept c_2 in O, their corresponding concepts c_1' and c_2' are related by the same relation. This leads the following definitions:

An Ontology o is a tuple O=(C, R, <, σ) with:

- C and Rare two disjoint sets called concept identifiers and relation identifiers *respectively*,
- partial order < on C called concept hierarchy or taxonomy
- function $\sigma : R \rightarrow C \times C$ called signature that associates a semantic relation to a couple of concepts.

An ontology *morphism* between two ontologies O=(C, R, <, σ)and O'=(C', R', <', σ') is the couple of functions (F,G) such that F : C → C' and G : R → R'. Given c and d two elements (concepts) of C and r element (relation) of R we note that:

F(c)=c' is the corresponding concept of c, F(d)=d' is the corresponding concept of d and G(r)=r' is an equivalent relation to r, particularly r' equal r.

Using ontology morphism, we can deduce these rules:

- *Rule1: If c < d then F(c) <' F(d); that means, If c precedes d in ontology O, then c'precedes d' in ontology O'*
- *Rule2: If σ(r)=(c,d) then σ'(G(r))=(F(c),F(d)); that means, if r is a relation between c and d in ontology O, then r(or r') is a relation between c' and d'*

B. Matchers Based on Semantic Relations

Semantic relations between concepts generated in the ontology enrichment step are exploited in the mapping process by specific matchers called semantic matchers. The semantic matchers do not require that the concepts of the two ontologies share the same set of resources. The principle of these matchers is that two concepts are more likely to be the same if their semantic neighborhood concepts are similar. We assume that the semantic neighborhood concepts, of a concept c,are a set of all concepts semantically related to it in the same ontology.

Thanks to the ontology *morphism* definition and after getting some mappings between concepts using linguistic, syntactic and structural matchers, semantic matchers produce new mappings or improve generated mappings between ontologies concepts, using the semantic relations generated in the enrichment step (Section III).

Semantic matchers provide mapping hypotheses represented by tuples in the form: <rel,c,d,$Conf_{SemMatcher}$, $Sv_{SemMatcher}$> where:

- rel: is a type of relation generated between the two concepts c and d. It can be the include relation (⊇), overlap relation (⊆) or equal relation (≡);
- $Conf_{SemMatcher}$: is the confidence level associated to the semantic matcher;
- $Sv_{SemMatcher}$: is the similarity value between c and d (comprise between 0 and 1) generated by the semantic matcher.

For each ontology concept, we calculate the cardinality of its semantic neighborhood. Figs. 4 and 5 illustrate two kinds of semantic neighborhood:

1. If the semantic relation whodefines this neighborhood is an anti-symmetry relation (Fig. 4), we find two different subsets: Set of semantic child and Set of semantic father. We associate to each set an index of cardinality: nbrOfSemChild calculates the cardinality of the semantic child set and nbrOfSemFather calculates the cardinality of the semantic father set.
2. If the semantic relation which defines this neighborhood is symmetry (Fig. 5), we have the only set regrouping all related concepts.

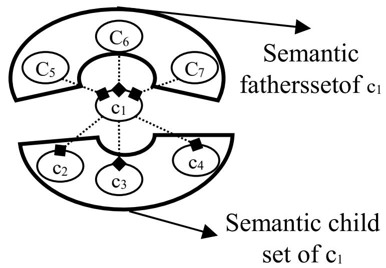


Fig. 4 Neighborhood of the anti-symmetry semantic relation

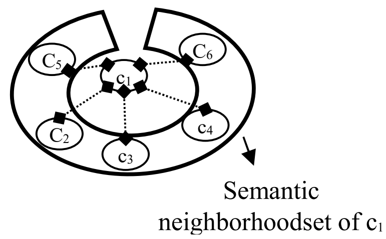
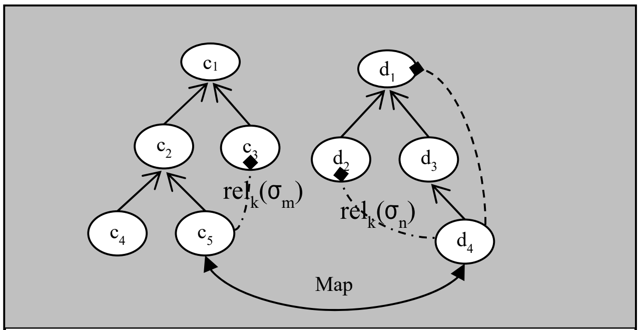


Fig. 5 Neighborhood of the symmetry semantic relation

The following examples of semantic matcher will illustrate the mechanism of similarity computation. Given the concepts c_i (respectively, d_i) of the ontology O_1 (respectively ontology O_2), and $rel_k(\sigma_n)$ (noted also, $rel_k(c_i,c_j,\sigma_n)$) represents the semantic relation between coupe of concepts c_i and c_j by σ_{relk}Jaccard measure.



The semantic relation $rel_k(\sigma_m)$ (e.g. strong-precede relation), is a anti-symmetry.
$rel_k(\sigma_m)$ links c_5 to c_3 in ontology1 and links d_4 to d_2 in ontology2.
NdrOfSemChild (c_i,$rel_k(\sigma_m)$) : is a number of semantic child concepts. E.g., NdrOfSemChild (d_4,$rel_k(\sigma_n)$)=2 (i.e., d_2 and d_1).

Matching Algorithm example

IF∃ Map<≡,c_5,d_4,$Conf_1$,SV_1>

THEN

$\sigma = Min(\sigma_n,\sigma_m)$;

MinOfSemChild(c_i,d_j)=Min (NdrOfSemChild(c_i,$rel_k(\sigma_n)$), NdrOfSemChild(d_j,$rel_k(\sigma_m)$))

IF∃Hp<≡,c_3,d_2,$Conf_2$,SV_2>

THEN

$SV_2 = SV_2 + \sigma*(SV_1/$ MinOfSemChild(c_5,d_4));

$Conf_2$=Conf2+ σ* $Conf_{SemMatcher}$;

ELSE generate Hp<≡,c_3,d_2,$Conf_2$,SV_2>

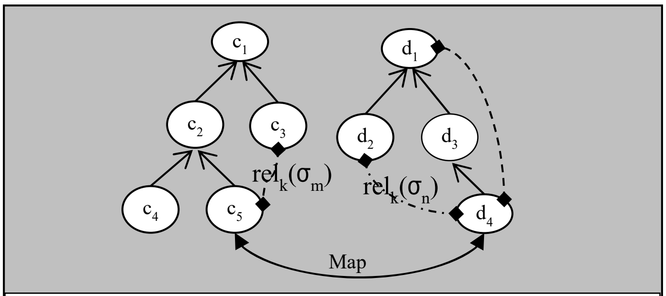
$SV_2 = SV_2 + \sigma*(SV_1/$ MinOfSemChild(c_5,d_4));

$Conf_2$=Conf2+ σ*$Conf_{SemMatcher}$;

Fig. 6 Semantic child matcher: using anti-symmetry relation

The anti-symmetry semantic relation is similar to structural relation; it defines for each concept a set of child (or sub-concept) and a set of father (or super-concept). The mapping between two concepts imply that exist a subset of their neighbourhood are similar. According to neighbourhood's type we define tree matchers: Semantic Child matcher (Fig. 6) when it considers the semantic child neighbourhood of an anti-symmetry relation; Semantic Father matcher when it considers the semantic father neighbourhood of an anti-symmetry relation and Semantic Concept Matchers when it considers the semantic neighbourhood of a symmetry relation (Fig. 7).

Another way to improve the matching process is to not only consider direct semantic relation but also the indirect one. The transitive characteristic allows defining this indirect semantic relation. Typically, if we have a transitive semantic relation which relates c_1 to c_2 and c_2 to c_3 we deduce the indirect semantic relation between c_1 and c_3.



The semantic relation $rel_k(\sigma_m)$ (e.g. strong-precede relation), is a symmetry relation.
$rel_k(\sigma_n)$ links d_1, d_2 and d_4 between them in ontology O_2.
NdrOfSemConcept(c_i,$rel_k(\sigma_m)$) : is a number of concepts $rel_k(\sigma_m)$ relation. E.g., NdrOfSemConcept(d_4,$rel_k(\sigma_n)$)=2

Matching Algorithm example

IF∃ Map<≡,c_5,d_4,$Conf_1$,SV_1>

THEN

$\sigma = Min(\sigma_n,\sigma_m)$;

MinOfSemConcept(c_i,d_j)=Min (MinOfSemConcept(c_i,$rel_k(\sigma_n)$), MinOfSemConcept(d_j,$rel_k(\sigma_m)$)

IF∃Hp<≡,c_3,d_2,$Conf_2$,SV_2>

THEN

$SV_2 = SV_2 + \sigma*(SV_1/$ MinOfSemConcept(c_5,d_4));

$Conf_2$=Conf2+ σ* $Conf_{SemMatcher}$;

ELSE generate Hp<≡,c_3,d_2,$Conf_2$,SV_2>

$SV_2 = SV_2 + \sigma*(SV_1/$ MinOfSemConcept(c_5,d_4));

$Conf_2$=Conf2+ σ*$Conf_{SemMatcher}$;

Fig. 7 Semantic neighborhood matcher: using symmetry relation

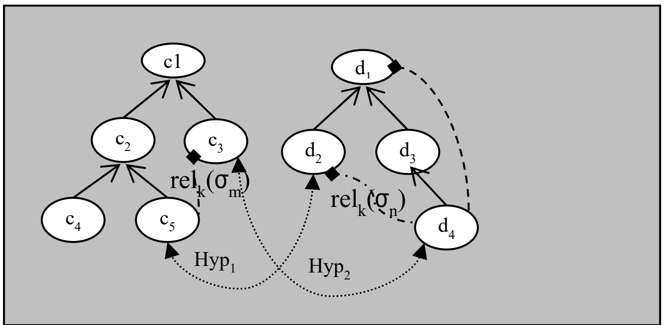
C. Filters Based on Semantic Relation

One of the characteristics of ROMIE is the ability to minimize the number of false candidate mappings and to help a user during the validation mapping process. We have developed several methods for filtering the mapping hypotheses generated. They are based on the structural and semantic relations of the ontologies. As matcher semantic, we exploit the semantic relations characteristics to eliminate the inadequate mapping automatically. The key idea is to respect the rules of ontology *morphism*. For example, Fig. 8 illustrates the principal of filtering when we have two contradictory hypotheses (i.e., if the hypotheses crosses), in this case we keep the strongly one. Like this filter we contribute another filter which verifies the compatibility between mapping hypotheses and existing mapping (i.e., the mapping hypotheses which already validated).

V. Prototype and Results

The prototype is implemented using the multi-agent system [6] with a platform JADE (Java Agent Development

Framework) [7]. We use OntoBroker system [8] to manage the ontologies. OntoBroker integrates various input formats of ontologies like RDF(S), F-Logic or OWL. The different matchers and similarity methods we developed are implemented with logic rules which make our system ROMIE easily extensible.



Hyp*i* is a hypothesis of mapping but it's not yet validated.
rel_k is a semantic relation which links c_5 to c_3 in the ontology O_1 and links d_4 to d_2 and d_1 in the ontology O_2.

Filtering Algorithm example

IFrel_k is an anti-symmetric relation And

$\exists Hp_1 <\subseteq, c_5, d_2, Conf_1, SV_1>$ and

$\exists Hp_2 <\subseteq, c_3, d_4, Conf_2, SV_2>$

THEN

IF $SV_1 > SV_2$

THEN Hp_2 is eliminated

ELSE Hp_1 is eliminated

Fig. 8 Filtering the crossed mapping hypotheses

We tested our system on an e-learning application. The following use case scenarios illustrate the main functionalities offered to the user as well as the communication between two ontologies a local one (LocOnto) and a distant one (DistOnto):

Scenario 1: a learner uses LocOnto as a support tool in her/his coursework. When trying to solve a specific task, however, she/he is not satisfied by the learning resource provided by LocOnto since it is not enough for her/him to achieve her/him learning goal. She/He announces this and the system (ROMIE) seeks external learning resource indexed by DistOnto. It sends a request to DistOnto for learning resources on the same topic (developing the same concept).

Scenario 2: an author uses LocOnto to construct a course. When preparing it, in addition to the learning resources she/he creates, she/he would like to reuse some available resources from DistOnto relevant to the course topics.

Requests are expressed by users on resources (instances). We could have for example the following queries:

- give me all available resources connected with the concept c;
- give me resources of kind K (e.g. using the LOM standard) connected to concept c;

Requests could also be expressed in terms of domain concepts and relationships between them. Below some examples of possible queries on ontology concepts:

- Give me the direct parents (or children) of concept c (i.e. the concepts in a direct 'is-a' relation with it);
- Give me all ascendants (or descendants) of concept c;
- Give me all concepts related (with a relation of any kind) to concept c.

We used two educative repositories where learning objects are indexed with ontologies of the educative domain: Simbad [4] and ACM [9].

- The *Simbad* ontology has been developed by our team; it annotates learning documents (courses, exercise, etc.).
- The *ACM/CCS* ontology is a classification for computer science domain. It classifies nine main sub-domains organized into sections.

In our tests, we consider a part of Simbad (30 concepts annotating 120 resources) as the local ontology (LocOnto) and a part of ACM (two sections: Computer systems organization section and software section, connected to 100 resources) as the distant ontology (DistOnto).

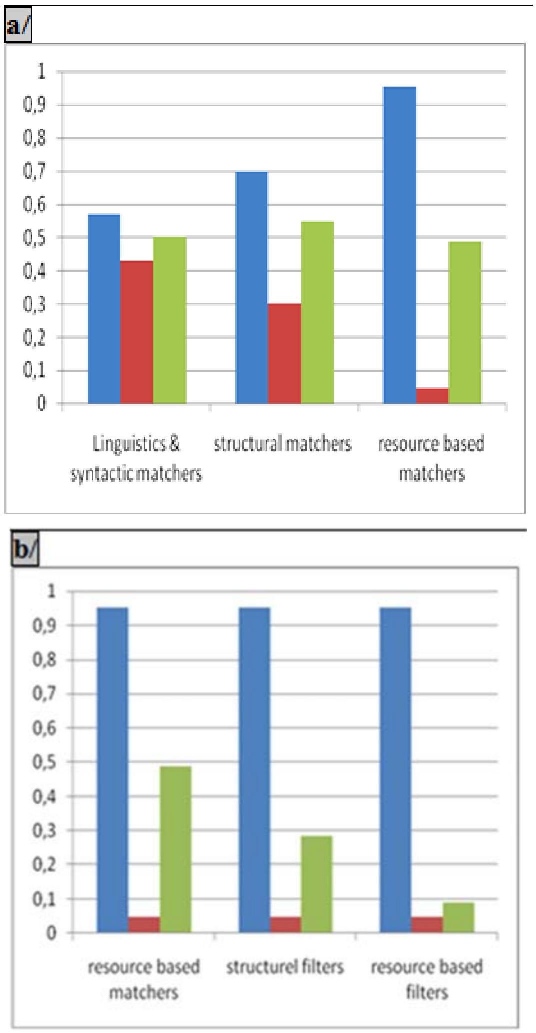
The two considered ontologies are in a first step automatically enriched by semantic relations thanks to the learning resources. The system offers the possibility to map directly the two ontologies otherwise the mapping is generated step by step during user queries in a transparent way. In this case, the mapping process is triggered only if there does not exist any resource in the local repository that satisfy user query or if the user is not satisfy by the query results.

We performed several tests in order to evaluate the performance of ROMIE in generating mappings between ontologies:

- In a first step, we analyze the impact of the resource based matchers (Fig. 9 (a)) on the mapping results. In this purpose, we first applied only linguistics and syntactic matchers then, in the second test we used the structural matchers to generate more mappings and in the last test we used also the semantic matchers. Each test is presented separately in order to thoroughly show the importance of each of the methods that ROMIE uses for generating mappings.
- In a second step, we analyze the impact of the resource based filters (Fig. 9 (b)) in improving the mapping results obtained in the first step. We firstly applied only the structural filters then we added the semantic filters, which use the semantic relations generated during ontology enrichment.

To evaluate the results obtained by ROMIE, we calculate three metrics for each result: percentage of true positive mappings (mappings correctly identified), percentage of false negative mappings (mappings not discovered) and percentage of false positive mappings (wrong mappings).

The mapping results are more and more improved thanks to the succession of methods of matching and filtering we use. We can see that when we use only linguistic and semantic matchers, we find more than 50% of the mappings, but 50% of the mappings generated by our system are false. Applying in addition the structural matchers, we obtain more mappings (70%) but more false mappings are also generated. Thanks to the resource based matchers (semantic matchers), we success to find almost all mappings (95%). Nevertheless, the number of wrong mappings still becomes high. But thanks to structural filters and to resource based filters (semantic filters), the number of wrong mappings decrease consequently, achieving some 10% on the total number of obtained mappings.



Correct mapping discovered
Mapping not discovered
Wrong mapping

Fig. 9 Mapping results obtained by ROMIE

VI. Related Work

A great number of ontology mapping approaches exist, as pointed out in[1]-[3]. All of them use several matchers, which are often of different types: syntactic, linguistic, structural (topological) and semantic. The semantic matchers are usually based on the semantic relations between the concepts in each of considered ontology. Some of existing approaches of ontology mapping considers the expressive ontology language for defining these relations. For example, in [10], authors use on a subset of OWL Lite in this purpose but they mainly focus on the comparison of the structural aspects of ontology. In our case, we enrich each of ontology by new semantic relations thanks to characteristics of resources connected to the ontology concepts. There are several works on ontology mapping based on the instance-based (resources-based) approach [5], [11], [12]. In all these works, to define a similarity measure between concepts, there is an explicit reference to the ontology model of OWL Lite and the similarity is defined among OWL objects (i.e., concepts) in terms of the number of common instances that characterize each concept.

In **Error! Unknown switch argument.**, [14], a system called *Automs* is proposed. It creates a semantic mapping based on ontology metadata. The ontology model adopted in this approach refers also to the hierarchy relation (IS-A). In[14], the proposed approach uses the Jaccard measure to calculate the statistics of common instances between two concepts. In[15], four matchers are defined in order to determine the instance-based similarity, using the number of instances that are associated or not associated to two compared concepts. In other words, the degree of similarity between two concepts takes into account the number of shared instances. In[16], authors address the problem of migrating instances between ontologies; they exploit existing mappings between ontologies to reclassify a set of instances of source ontology into related target ontology.

Almost existing instance based approaches assume that the instance level is shared between ontologies. Therefore, the instance-based mapping between concepts of two ontologies is determined based on the overlap of their instance sets. In our case, we have two different instance levels (of one each ontology) and the purpose is to generate and exploit the generated concept mappings in order to migrate and exchange resources between ontologies (or repositories). Another important difference between ROMIEE and existing instance-based approaches is the use of the resources and the ontology structure to determine automatically the wrong generated mapping (filtering step). In fact, the filtering process in existing works is generally based only on threshold value.

VII. Conclusion

Thanks to the web, users could access to a huge amount of data and resources. In a large number of domains, such as e-learning, biology, and so on, ontologies are more and more used to allow the exchange of these data and resources between different users. Unfortunately, there is rarely one of global ontology in a given domain. Several ontologies must be considered, that's why the task of mapping ontologies is very important. In this paper, we have presented a system called ROMIE and an algorithm which allow in one hand to search for distant resources requested by users thanks to ontology mapping and in the other hand to perform ontology mapping thanks to user queries. We have therefore proposed a mapping algorithm based on resources which improves consequently the mapping process. The principle is to enrich, in a first step, the ontology by semantic relations between its concepts. We have applied ROMIE on an e-learning application, and the obtained results are very promising. A further work is to test ROMIE on a biology application, which will require necessarily adaptations of the system. Moreover, the size of ontologies used in biology is often high and it will be interesting to test ROMIE with a large number of concepts and resources. Another way to improve the matching process is to not only consider direct semantic relation but also the indirect one. The transitive characteristics allow defining this indirect semantic relation. Typically, if we have a transitive semantic relation which relates c_1 to c_2 and c_2 to c_3 we deduce the indirect semantic relation between c_1 and c_3.

Finally, a battery of tests with real end users is planned. Their feedback will be very useful to improve the results. Some of the feedbacks will be captured automatically.

References

[1] Euzenat, J., Shvaiko, P. Ontology matching: state of the art and future challenges (2nd ed.). Heidelberg (DE): Springer-Verlag. URL:http://book.ontologymatching.org (2013).

[2] Otero-Cerdeira, L., Rodreguez-Martinez, F. J. & Gomez-Rodriguez A.: Ontology matching: A literature review in a literature reviewExpert Systems with Applications, (2015).

[3] Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng-Kuntz, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt H., Shvaiko, P., Tessaris, S., Van Acker, S., Zaihrayeu, I.: State of the art on ontology (2004).

[4] Bouzeghoub, A., Defude, B., Duitama, F. &Lecocq, C. (2006). /A Knowledge-Based Approach to Describe and Adapt Learning Objects/. International Journal on E-Learning. 5 (1*) *Special Issue: Learning Objects in Context, pp. 95-102. Chesapeake, VA: AACE (ISSN 1537-2456).

[5] Bouzeghoub, A., Elbyed, A. Ontology Mapping for Web-Based Educational Systems Interoperability. IBIS (Interoperability in Business Information Systems), vol. 1, 73-84 (2006).

[6] Bouzeghoub, A., Elbyed, An ontology mapping algorithm to share learning resources. Information and Communication Technologies, 2006. ICTTA'06. 2nd 1, 616-621

[7] Java Agent Development Framework, http://jade.tilab.com/

[8] OntoBroker user guide, http://www.ontoprise.de/

[9] ACM Computing Classification System: Advancing Computing as a Science & Profession: available from http://www.acm.org/

[10] Euzenat, J. and Valtchev, P.: Similarity-Based Ontology Alignment in OWL-Lite. ECAI. Valencia, Spain IOS Press, 333-337 (2004)

[11] Albertoni, R., De Martino, M.:Semantic Similarity of Ontology Instances Tailored on the Application Context.In: Proceedings of ODBASE (2006).

[12] Bernstein, P., Melnik, S., Mork, P.:Interactive Schema Translation with Instance-Level Mappings.In: Proceedings of VLDB (Demonstration) (2005)

[13] Kotis, K., Valarakos, A., Vouros, G.: AUTOMS: Automating Ontology Mapping through Synthesis of Methods. OAEI (Ontology Alignment Evaluation Initiative) contest, Ontology Matching International Workshop, Atlanta, USA. (2006)

[14] Valarakos, A. G., Spiliopoulos, V., Kotis, K., Vouros G.:AUTOMS-F: A Java Framework for Synthesizing Ontology Mapping Methods. In: Proceedings of I-KNOW (2007).

[15] Kirsten, T., Thor, A., Rahm, E.: Instance-based matching of large life science ontologies.In: Proc. 4th Int. Workshop DILS(Data Integration in Life Sciences), LNCS, 172-187 (2007)

[16] Serafini, L., Tamilin, A.:Instance Migration in Heterogeneous Ontology Environments. In: Proceedings of ISWC+ASWC (2007)

A. Elbyed was born in Casablanca- Morocco on April 17, 1978 He is a professor in department of Mathematics and Computer science, at Hassan II University, faculty of sciences Ain Chock, Casablanca - Morocco. He obtained a PhD in the National Institute of Telecommunications (INT) of Evry, France, where he participated in European projects, such as *SAPHIRE*and *POPS*. His research focuses on semantic web and the automatic detection of contextual similarity (Ontology Mapping) using artificial intelligence. He is the author or co-author of several articles published in journals and international conferences (IEEE).