Taking Advantages of Ontology and Contexts to Determine Similarity of Data*

Agustina Buccella, Alejandra Cechich
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue, Neuquén, Argentina
{abuccell, acechich}@uncoma.edu.ar

and

Nieves Rodríguez Brisaboa
Departamento de Computación
Universidade de A Coruña, España
brisaboa@udc.es

Abstract
Data integration is the process of unifying data sharing some common semantics but are originated from unrelated sources. In our work we consider these sources are autonomous, heterogeneous and they are physically distributed. These three characteristics make the integration task more difficult as there are several aspects to bear in mind. In this work we only focus on one of these aspects, the semantic heterogeneity, which deals with the meaning of the concepts within the information sources. As each source contains a specific vocabulary according to its understanding of the world, terms denoting same meaning can be very difficult to find. In this paper we will briefly explain our method to find similarities using ontologies and contexts. We will propose some improvements in the similarity functions in order to take advantages of the information the ontologies provide.
Keywords: Semantic Heterogeneity, Ontology, Context, Similarity

1. Introduction

The semantic heterogeneity is one of the most complex problems within data integration tasks. Each information source included in the integration has its own interpretation and assumptions about the concepts involved in the domain. Therefore, it is very difficult to determine when two concepts belonging to different sources are related. Some relations among concepts that semantic heterogeneity involves are: synonymous, when the sources use different terms to refer to the same concept; homonymous, when the sources use the same term to denote completely different concepts; hyponym, when one source contains a term less general than another in another source; and hypernym, when one source contains a term more general than another in another source; etc.

In order to deal with some semantic heterogeneity problems, in recent works [4,5,6,7] we have proposed the combination of two useful tools: ontologies and contexts. In our works an ontology includes the conceptual vocabulary (terms and relationships) and the rules and axioms relating terms within the vocabulary.

This work is partially supported by the CyTED (Ciencia y Tecnología para el Desarrollo) project VII-J-RITOS2, and the UNComa project 04/E048 (Modelado de Componentes Distribuidos Orientados a Objetos).

According to [23] an ontology is defined as a 5-tuple O =< C,R,F,I,A > in which C is a set of classes, R a set of relations, F a set of functions, I a set of instances, and A a set of axioms. This definition is based on the Ontolingua language specification [15], however in this paper, we will use the Web Ontology Language (OWL) [21,1] to represent the ontologies due to its widespread use in the Semantic Web [3]. So, the 5-tuple definition is reduced to 3-tuple O =<C,P,R> in which C is a set of classes as before, P is a set of properties and R is a set of restrictions applied to the classes and properties.

On the other hand, the contexts are useful to model concepts which are in conflict with one another, that is, a concept can vary its meaning according to the context it is in. In our works, a context is a set of classes and properties indicating a specific role of the database, which provides semantic knowledge that can be used to integrate several data sources. For example, the sold cars can be a context involving the car and buyer classes and the buy property. The use cases of a UML specification [10] might be the source to obtain some of the contexts.

In [5] we have described our method especially built to integrate different ontologies. Besides, an approach based on the hybrid ontology approach [24] has been defined in order to simplify the integration. Figure 1 shows this approach. As we can see, one source ontology is built for each information source. The OCM (Ontology and Context Mapping) component deals with the relationships among the contexts and concepts of the different source ontologies, and with the information flow between the source ontologies and the shared vocabulary. The shared vocabulary component is composed of the generic concepts and contexts that will be used to query the system.

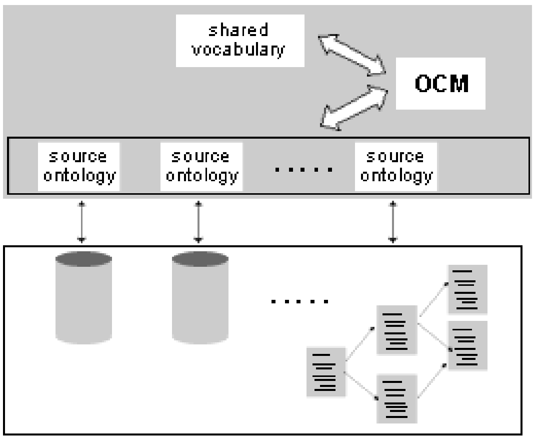

Figure 1. Our approach to data integration

The construction of each component is based on a method with three main stages (Figure 2): *building the source ontology*, *building the mappings between source ontology and shared vocabulary* and *modifying the OCM and shared vocabulary.* This method proposes a guide in order to do this activity more consistent and correct. To achieve each stage, a set of steps should be performed. We briefly explain each stage in order to make the process clearer.

The first stage, building the source ontology, contains three main steps: *generating the OWL initial ontology*, *adding semantics* and *defining contexts.*

The first step, *generating the OWL initial ontology*, takes as input an Entity-Relation model (ER) and a relational model [8,9] and automatically generates an initial ontology. In order to do the initial ontology, we use the semantic information provided by the ER model. By a series of rules we transform this model into an ontology. The ontology will be represented by using OWL [21,1]. As we have indicated before, in our examples, we have chosen OWL due to its widespread use in the Semantic Web [3]. Besides, OWL allows formalizing a domain by defining classes and properties of those classes, defining individual's asserting properties, and reasoning about these classes and individuals to the degree permitted by the formal semantics of the OWL language. OWL can be (partially) mapped to a description logic [2] making possible the use of existing reasoners such as FACT [17] and RACER [16].
The second step, *adding semantics*, allows the expert (for example, using an ontology editor as Protégé [12] with the OWL plug-in) to add restrictions, classes and/or properties to the initial ontology. Knowing the domain of the information source and understanding the structures, the user is able to provide more semantics to the ontology. Finally, the last step, *defining contexts*, implies the definition of the several contexts within an ontology. Besides, the expert user will have to determine the concepts that are included in these contexts.

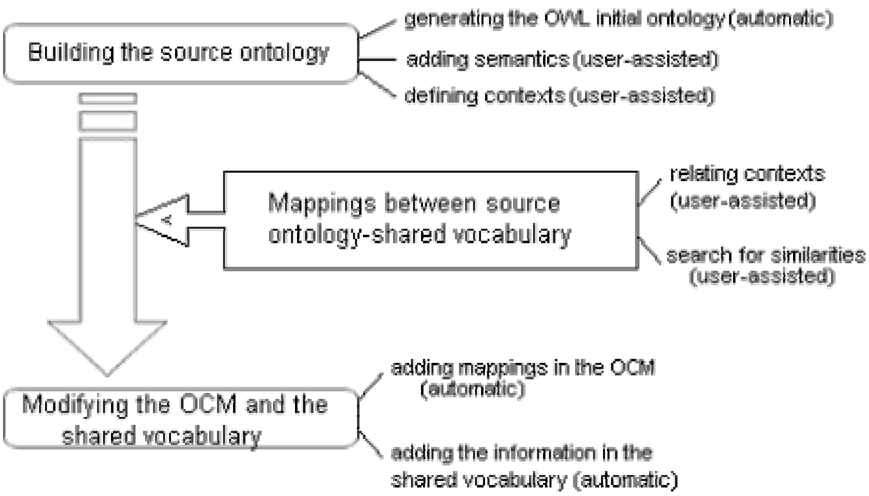

Figure 2. A method for data integration

The second stage, *building the mappings between the source ontology and the shared vocabulary*, contains two main steps: *relating contexts* and *searching for similarities*.
The first step implies defining the relationships between the contexts of the source ontology, built in the previous stage, and the shared vocabulary. The shared vocabulary has already defined its contexts, and the expert user must relate them with the user-defined contexts for the source ontology.
The second step, *search for similarities*, implies searching for similarities between the related contexts. In previous works [4,5] we have shown the use of two similarity functions proposed in [19,20] in order to find similarities among the classes and properties included in the related contexts.

The third stage, *modifying the OCM and the shared vocabulary*, contains two main steps: *adding mapping in the OCM* and *adding the new information in the shared vocabulary*. Both are automatic steps, that is, we will create a system that implements these steps without user intervention.
The former step is achieved by using the similarities found in the last stage. The OWL ontology mapping constructors will be used to store the mappings in the OCM component.
The latter step, *adding the new information in the shared vocabulary*, adds the information the shared vocabulary does not contain but it is provided by the source ontology. Thus, the shared vocabulary will make available all the information the sources ontologies offer.

Focusing on the second stage, *building the mappings among source ontologies*, we will divide this paper into two sections. Section 2 will explain in detail this stage and the similarity functions used in the process. These similarity functions have been defined by Rodriguez & Egenhofer in [19,20] and we have applied them to our method. Other related works can be found in [11,13,14]. For example, in [11] the similarity measure is not defined directly by a formula. Rather, it is derived from a set of assumptions about similarity. Another example can be found in [14] where the context becomes important for similarity assessment, because it affects the determination of the relevant features. The main advantage of the formulas presented in [19,20] is the combination of two different approaches to similarity assessment – the feature-matching process and the semantic distance. However, there is information required by the formulas that is not easy to obtain from the ontologies.

In section 3, we will show our proposal to improve the Rodriguez & Egenhofer's formulas in order to take advantages of the semantic information provided by the ontologies. Conclusions and future work will appear in the last section.

2. Searching Similarities

The Ontology and Context Mapping (OCM) component makes the mapping between the source ontologies and the shared vocabulary. To do so, the OCM relates the contexts defined in one ontology with the contexts defined in other ontologies. These relationships can be equality, inclusion, intersection, etc. Each context contains a set of concepts. Only the concepts included in the related contexts will be compared. In this way, we avoid comparing every concept in one ontology with every concept in one another. For instance, Figure 3 shows a part of two ontologies about animals using the OWL syntax.

Ontology 1 defines a *has_part* property relating the *Animal* class with the *Organ* class. As the *Animal* class is the domain and the *Organ* class is the range of the property, it indicates the organs of an animal. Besides, the *Animal* and *Organ* Classes are also defined and particularly the *Animal* class has a minimal cardinality restriction indicating that the animal has at least one organ. Ontology 2 has similar classes and properties, but it represents the *has_part* property with a *Creature* class.

Next, as we can see, in Figure 3, each ontology has defined one specific context: c_{11} and c_{21}. These contexts represent a specific query within the databases and include the concepts showed in the figure. The concepts included in these contexts will be compared between them. For example, we can relate the contexts as follow:

$(O_1,Context_1)\ (c_{11}) = (O_2,Context_2)\ (c_{21})$

In previous works [5,6], to find similarity values between two concepts within related contexts, we have used the similarity functions defined in [19,20]. These functions are useful to assess similarity because the concepts are analyzed in terms of their distinguished features together with their semantic relations.

PART OF THE ONTOLOGY 1	**PART OF THE ONTOLOGY 2**

`<owl:Class rdf:ID="`**`Animal`**`">`
`<rdfs:subClassOf>`
`<owl:Restriction>`
`<owl:onProperty rdf:resource="#has_part"/>`
`<owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"`
`>1</owl:minCardinality>`
`</owl:Restriction>`
`</rdfs:subClassOf>`
`<rdfs:comment>A living creature, not a plant</rdfs:comment>`
`</owl:Class>`
 	
`<owl:Class rdf:ID="`**`Organ`**`">`
`<rdfs:comment>A part of an animal that has a special purpose</rdfs:comment>`
`</owl:Class>`
 	
`<owl:ObjectProperty rdf:ID="`**`has_part`**`">`
`<rdfs:comment>The organs of an animal</rdfs:comment>`
`<rdfs:domain rdf:resource="#Animal"/>`
`<rdfs:range rdf:resource="#Organ"/>`
`</owl:ObjectProperty>`
	`<owl:Class rdf:ID="`**`Creature`**`">`
	`<rdfs:subClassOf>`
	`<owl:Restriction>`
	`<owl:onProperty rdf:resource="#has_part"/>`
	`<owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"`
	`>1</owl:minCardinality>`
	`</owl:Restriction>`
	`</rdfs:subClassOf>`
	`<rdfs:comment>A living creature, not a plant</rdfs:comment>`
	`</owl:Class>`
	
	`<owl:Class rdf:ID="`**`Organ`**`">`
	`<rdfs:comment>A part of an creature that has a special purpose</rdfs:comment>`
	`</owl:Class>`
	
	`<owl:ObjectProperty rdf:ID="`**`has_part`**`">`
	`<rdfs:comment>The organs of a creature</rdfs:comment>`
	`<rdfs:domain rdf:resource="#Creature"/>`
	`<rdfs:range rdf:resource="#Organ"/>`
	`</owl:ObjectProperty>`

CONTEXTS	**CONTEXTS**
$Ontology_1 = O_1$ $Context_1$ = animal_planet c_{11} = organs_of_an_animal …….. (O_1,$Context_1$) (c_{11}) = { animal, organ, has_part} ………………….	$Ontology_2 = O_2$ $Context_2$ = creature_planet c_{21} = organs_of_a_creature …….. (O_2,$Context_2$) (c_{21}) = { creature, organ, has_part} ………………….

Figure 3. Part of the two ontologies

The (1) and (2) functions show the formulas where a and b are concepts of two ontologies (O_1 and O_2 respectively).

$$S(a^{O_1}, b^{O_2}) = w_p.S_p(a^{O_1}, b^{O_2}) + w_f.S_f(a^{O_1}, b^{O_2}) + w_a S_a(a^{O_1}, b^{O_2}) \quad for \quad w_p, w_f, w_a \geq 0 \;\; and \;\; w_p + w_f + w_a = 1 \tag{1}$$

$$S(a,b) = \frac{|A \cap B|}{|A \cap B| + a(a,b)|A \setminus B| + (1 - a(a,b))|B \setminus A|} \quad for \quad 0 \leq a \leq 1 \tag{2}$$

The function (1) is a sum of products (value times weight (w)) where w represents the parts, the functions and the attributes (w_p, w_f, and w_a respectively). This model is called feature matching, where parts (S_p) are structural elements of a concept (or class), such as roof and floor of a building, function (S_f) represent the purpose of the concept, and attributes (S_a) correspond to additional characteristics of a concept. The function (2) is based on the Tversky's model [22] where A and B correspond to description sets of a and b (i.e., synonym sets, sets of distinguishing features, etc). The parts, functions and attributes are compared using this function. For example, if we compare the parts, $|A \cap B|$ represents the amount of equal parts between two concepts, $|A \setminus B|$ represents the amount of parts of A that are not in B and $\alpha(a,b)$ is a function that calculate the depth of the concept in a hierarchy. We refer the reader to [5] for more details.

Now, if we compare the animal concept in the Ontology 1 with the creature concept in the Ontology 2, the similarity values will be high because the parts, the function and the attributes are very similar. Figure 4 shows these elements for the *Animal* class and the *Creature* class of our example. We have used WordNet [18] in order to obtain the parts and functions of each concept.

If we applied the functions (1) and (2) to these classes, we obtain the following values, because all the elements are the same:

$$S_p(animal, creature) = 1, \quad S_f(animal, creature) = 1 \;\; and$$

$$S_a(animal, creature) = 1 \;\; \Rightarrow \;\; S(animal^{O1}, creature^{O2}) = 1$$

As we can see, we have compared these concepts without thinking of the underlying ontologies because the parts, functions and attributes are difficult to obtain from the ontologies. For example, the parts of a thing (class) in an ontology can be divided into several properties as well as attributes. Besides, there are abstract classes and properties that do not have parts. So, these similarity functions only capture some of the information the ontologies provide. For example, the instances of a concept are not represented, an attribute can be also a class with more attributes and so on, etc.

CLASS ANIMAL	CLASS CREATURE
Parts = {eyes, organ, mouth, … } Function = {a living organism characterized by voluntary movement} Attributes = {age, weight, ...}	Parts = {eyes, organ, mouth ,…} Function = {a living organism characterized by voluntary movement} Attributes = { age, weight, ...}

Figure 4. Parts, functions and attributes of two classes

In this paper, we will propose a guideline to recover all the information the ontologies provide without the need of consulting other sources. These sources as WordNet are used to obtain information, which is very difficult to extract from the ontologies. In the next section we will introduce changes on these similarity functions in order to take advantages of the ontologies and make the use of all data easier.

3. Improvements to the Similarity Functions

The similarity functions (1) and (2) described above, compare two concepts using their parts, functions and attributes. Besides these three elements, an ontology contains more information. For instance, one ontology may have the parts (in the way of relations or functions) of the car concept, but also other properties such as capacity, owner, etc, which can be represented by other attributes.

The extraction of attributes within an ontology can be a hard task because it is very difficult to determine which elements are attributes and which correspond to the parts of the concept. Therefore, we propose to modify the function (1) in order to use all the necessary information. Function (2) will remain unchanged and it will be used by all the next functions.

To make an easier comparison among the concepts, Figure 5 shows how the different elements of an ontology are divided. When we talk about concepts, we refer to any element of the ontology.

The first division in Figure 4 refers to the comparison of different elements. On one branch, we have the classes and on the other branch, the properties.

Firstly we analyze the class branch, which is divided into two new branches: common classes and attribute classes. Both are classes defined in the ontology to represent things about the world. A class is a unary relation, a set of tuples (lists) of length one. Each tuple contains an object which is said to be an instance of the class. An individual, or object is any identifiable entity in the universe of discourse, including classes themselves. The specific role defined in the ontology is the difference between them. The common classes have the role of representing things about the domain and the attribute classes have the role of representing information about a common class (attribute). Both roles exist because the ontologies do not have the concept of attribute. For example, if an information source has represented the datum name as a string of 20 characters, this name is an object in the ontology with many properties, it is not a string because the classes and object are about the world and not about data structures.

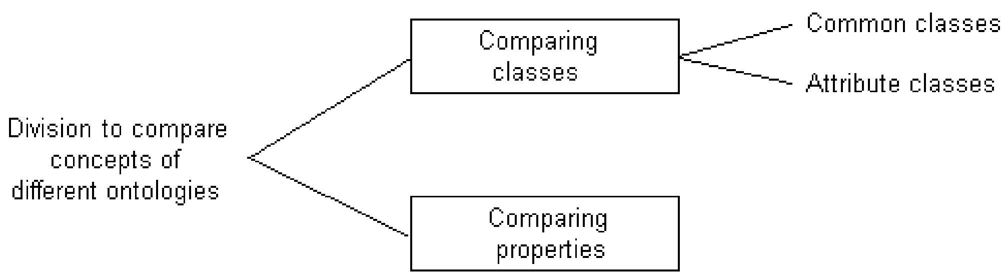

Figure 5. Proposed division to compare concepts

On the other branch, Figure 5 shows properties. A property is a set of tuples that represents a relationship among objects in the universe of discourse. Each tuple is a finite, ordered sequence (i.e., list) of objects. A property is also an object itself, namely, the set of tuples.

Figure 6 shows the example of the Ontology 1 defined in OWL with classes and relations, and it specifies the common classes and the attribute classes according to the way they are defined. The *has_part* property is enough to classify organ as an attribute class because it denotes a property about the animal class. Then, according to this classification, we can define the similarity functions for each element in Figure 5. Given two related contexts, the common classes within one context (of one ontology) and the common classes within another context (of another ontology) will be compared using the similarity function:

$$S(a^{O_1}, b^{O_2}) = w_i.S_i(a^{O_1}, b^{O_2}) + w_e.S_e(a^{O_1}, b^{O_2}) + w_a S_a(a^{O_1}, b^{O_2}) \quad for \quad w_i, w_e, w_a \geq 0 \quad and \quad w_i + w_e + w_a = 1 \tag{3}$$

Similarly to the function (1), the function (3) is a sum of products (value times weight (w)) where w represents the individuals, the explanation and the attributes (w_i, w_e, and w_a respectively). The individuals (S_i) are objects of a class/es. An object is an individual of a class if it is a member of the set denoted by that class. The explanation (S_e) is the comment part expressed in natural language within the OWL definition. For example, a living creature not a plant is the explanation of the animal concept. The attributes (S_a) correspond to additional characteristics of a concept.

PART OF THE ONTOLOGY 1

```
<owl:Class rdf:ID="Animal">
   <rdfs:subClassOf>
    <owl:Restriction>
     <owl:onProperty rdf:resource="#has_part"/>
     <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
     >1</owl:minCardinality>
    </owl:Restriction>
   </rdfs:subClassOf>
  <rdfs:comment>A living creature, not a
plant</rdfs:comment>
</owl:Class>
```

Animal plays the role of a common class because it defines a thing about the universe of discourse.

```
<owl:Class rdf:ID="Organ">
  <rdfs:comment>A part of an animal that has a special
purpose</rdfs:comment>
</owl:Class>
```

Organ plays the role of an attribute class because it exists to describe a characteristic about a common class (animal).

```
<owl:ObjectProperty rdf:ID="has_part">
  <rdfs:comment>The organs of an animal</rdfs:comment>
   <rdfs:domain rdf:resource="#Animal"/>
   <rdfs:range rdf:resource="#Organ"/>
</owl:ObjectProperty>
```

has_part is the property denoting that organ plays the role of an attribute class.

Figure 6. An example of a common and attribute class and a relation

In our example, we should use the function (3) if we want to compare the *Animal* and *Creature* classes. Figure 7 shows the individuals of the *Animal* and *Organ* classes and *Creature* and *Organ* classes, for Ontology 1 and Ontology 2 respectively. These individuals are necessary to calculate the similarity values.

Note that the *Animal* class in the Ontology 1 contains both mammal and not mammal animals. But the *Creature* class in the Ontology 2 only contains not mammal animals. This difference is not taken into account in the similarity functions of Section 2 because the individuals are not involved. Therefore, when we apply the function (3) we will obtain different values.

Firstly, we will apply the function (2) to the attributes. In this case, for both ontologies the only attribute is the *Organ* class. Then the function is:

$$S_a(animal, creature) = \frac{1}{1 + \alpha.0 + (1-\alpha).0} = 1$$

where α is equal to 0.5 because both classes are in the same hierarchical level (see [19,20] for more details).

The function applied to the explanation is also equal to 1, $S_e(animal, creature) = 1$

But the function applied to the individuals will return a different value because the individuals are different:

$$S_i(animal, creature) = \frac{2}{2 + \alpha.2 + (1-\alpha).0} = 0.67$$

Then, the function (3) returns the following value:

$$S(animal^{O_1}, creature^{O_2}) = ((w_a)x1) + ((w_e)x1) + ((w_i)x(0.67)) = 0.88$$

This result denotes that the *Animal* and *Creature* classes are not exactly equal as the function (1) indicated in the previous section. This new value is consistent with the domains because the individuals in this case denote a little difference in the meaning of the concepts. In this way we make use of all the information the ontologies provide.

INDIVIDUALS OF THE ONTOLOGY 1	**INDIVIDUALS OF THE ONTOLOGY 2**
<Organ rdf:ID="mouth"/> <Organ rdf:ID="udder"/> <Organ rdf:ID="eyes"/> <Animal rdf:ID="fish"> <has_part rdf:resource="eyes"/> <has_part rdf:resource="mouth"/> </Animal> <Animal rdf:ID="bird"> <has_part rdf:resource="eyes"/> </Animal> <Animal rdf:ID="dog"> <has_part rdf:resource="mouth"/> <has_part rdf:resource="eyes"/> <has_part rdf:resource="udder"/> </Animal> <Animal rdf:ID="cat"> <has_part rdf:resource="mouth"/> <has_part rdf:resource="udder"/> <has_part rdf:resource="eyes"/> </Animal>	<Organ rdf:ID="mouth"/> <Organ rdf:ID="eyes"/> <Creature rdf:ID="fish"> <has_part rdf:resource="eyes"/> <has_part rdf:resource="mouth"/> </Creature> <Creature rdf:ID="bird"> <has_part rdf:resource="eyes"/> </Creature>

Figure 7. Some individuals of the two ontologies

Now, for the attributes and given two related contexts, the attribute classes within one context (of one ontology) and the attribute classes within another context (of another ontology) will be compared. But we will only compare attribute classes of common classes that have already been compared (in the previous step) and have obtained a high similarity value. The similarity function used is the same; only the attribute product is deleted because these types of classes do not have attributes. Note that the function (1) does not provide a way of comparing attribute classes. The attribute does not contain parts or other attributes. Therefore, our similarity function for the attributes is:

$$S(a^{O_1}, b^{O_2}) = w_i.S_i(a^{O_1}, b^{O_2}) + w_e.S_e(a^{O_1}, b^{O_2}) \quad for\ w_i, w_e \geq 0\ and\ w_i + w_e = 1 \quad (4)$$

This function must be applied to the *Organ* classes of the two ontologies because they do not have attributes. So, the similarity values are:

$$S_e(organ^{O_1}, organ^{O_2}) = 1,$$
$$S_i(organ^{O_1}, organ^{O_2}) = \frac{1}{1 + a.1 + (1 - a).0} = 0.67 \Rightarrow$$
$$S(organ^{O_1}, organ^{O_2}) = 0.83$$

If any class in one ontology plays the two roles (common class and attribute) the function (3) should be used because the $w_a.S_a(a^{O_1}, b^{O_2})$ will be different from zero.

Finally, another change in the similarity function is applied when we need to compare properties. The function (1) cannot compare properties as a whole because there is no way to put the domain and range together in the same function. Therefore, the similarity function applied to the attributes will be replaced by the comparison of the domain and range into two separate values. Probably, the domain and range classes have been compared previously and we only must use the resultant values, and not recalculate them. The similarity function applied to the explanation and to the individuals does not change. Therefore, our similarity function for the properties is:

$$S(a^{O_1}, b^{O_2}) = w_d.S(domain(a)^{O_1}, domain(b)^{O_2}) + w_r.S(range(a)^{O_1}, range(b)^{O_2}) + w_i S_i(a^{O_1}, b^{O_2}) + w_e S_e(a^{O_1}, b^{O_2})$$
$$for\ w_d + w_r = 0.5\ and\ w_i + w_e = 0.5 \quad (5)$$

This function must be applied to the *has_part* properties of both ontologies. If we assume that the individuals of these properties are the logical combination between the *Animal* (or creature) classes and the *Organ* classes (for example, one individual can be Dog-Mammals for the Ontology 1, another Cat-Mammals, etc.) as Figure 7 shows, we obtain the following similarity values:

$$S(has\_part^{O_1}, has\_part^{O_2}) = w_d.S(animal^{O_1}, creature^{O_2}) + w_r.S(organ^{O_1}, organ^{O_2}) +$$
$$w_i.S_i(has\_part^{O_1}, has\_part^{O_2}) + w_e.S_e(has\_part^{O_1}, has\_part^{O_2}) = w_d.0.88 + w_r.0.83 + w_i.0.5 + w_e.1 = 0.80$$

Note that the w values are used to give more o less weight to the different elements within the functions. For simplicity, in this paper we have always used the same weight values. But in some situations changing these values can be useful. For example, a low value of the individual weight (w_i) would be more appropriated when domains are different.

4. Conclusion and Future Work

Based on the use of ontologies and contexts we have briefly presented an introduction of our approach to deal with semantic heterogeneity problems. We have focused on the second stage of our method in which the process to find similarities is done. Firstly, we have shown the application of the similarity functions defined by Rodriguez & Egenhofer using WordNet as another source of information, because recovering this information from the ontologies is very difficult. Then, we performed two new tasks: classifying the elements of an ontology and making improvements in the similarity functions. This classification is useful to retrieve the information the ontologies provide and to use it in the improved similarity functions. Also, we have created new similarity functions based on the previous functions in order to compare more detailed components such as classes and relations.

Our proposal depends exclusively on the information provided by the ontologies generating incorrect results when we work with incomplete ontologies or when some information is not used in the formulas. Note that our example defines two ontologies with enough information to find a minimum difference between the concepts. We include the whole information in the formulas.

As future work, we are building an automated tool to assist in the similarity process including all the information. Besides, we will build empirical proofs to show that our formulas are more effective in the search of similarity process. Finally, the approach and their extensions need be validated by using more complex examples and real cases for study.

References

1. Antoniou, G., Harmelen F. *Web Ontology Language: OWL. Handbook on Ontologies in Information Systems.* Staab & Studer Editors. Springer-Verlag, 2003.
2. Baader, F., Calvanese, D., McGuiness, D., Nardi, D. and Patel-Schneider, P. editors. *The Description Logic Handbook-Theory, Implementation and Applications*. Cambridge University Press, ISBN 0-521-78176-0, 2003.
3. Berners-Lee, T. *Weaving the Web*. Texere Publishing Ltd. ISBN: 0752820907. June 2001.
4. Buccella A., Cechich A. and Brisaboa N.R. An Ontology Approach to Data Integration. *Journal of Computer Science and Technology. Vol.3*(2). Available at http://journal.info.unlp.edu.ar/default.html, pp. 62-68, 2003.
5. Buccella A., Cechich A. and Brisaboa N.R. An Ontological Approach to Federated Data Integration. *9° Congreso Argentino en Ciencias de la Computación, CACIC'2003*, La Plata, October 6-10, pp. 905-916, 2003,.
6. Buccella A., Cechich A. and Brisaboa N.R. A Context-Based Ontology Approach to Solve Explanation Mismatches. *Jornadas Chilenas de Computación. JCC 2003*. Chillán, Chile, November 3-9, 2003.
7. Buccella A., Cechich A. and Brisaboa N.R. An Ontology-based Environment to Data Integration. *VII Workshop Iberoamericano de Ingeniería de Requisitos y Desarrollo de Ambientes de Software*. IDEAS 2004, pp. 79-90, 3-7 May, 2004.
8. Chen, P. The Entity-Relation model- Toward a unified view of data. *ACM Transaction on database systems*, Vol.1(1), pp. 9-36, March 1976.
9. Codd, E. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, Vol.13(6), pp. 377-387, 1970.
10. Fowler, M. and Scott, K. *UML distilled*, Addison-Wesley 1997.
11. Lin, D. An Information-Theoretic Definition of Similarity. *Int'l Conf. Machine Learning (ICML'98)*, 1998.
12. Gennari, J., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., Tu, S. W. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. Technical Report, SMI-2002-0943, 2002.
13. Goldstone, R. Similarity, Interactive Activation and Mapping. *Journal Experimental Psychology: Learning, Memory and Cognition. Vol. 20*, pp. 3-28, 1994.
14. Goldstone, R., Medin, D., Haberstadt, J. Similarity in Context. *Memory and Cognition, Vol. 25*(2), pp.237-255, 1997.
15. Gruber T. Ontolingua: A Mechanism to Support Portable Ontologies. Knowledge Systems Laboratory, *Stanford University*, Stanford, CA, Technical Report KSL 91-66, 1992.

16. Haarslev, V. and Moller, R. RACER system description. In P. Lambrix, A. Borgida, M. Lenzerini, R. Moller, and P. Patel-Schneider, editors, Proceedings of the International Workshop on Description Logics, number 22 in CEUR-WS, Linkoeping, Sweden, July 30-August 1 1999.
17. Horrocks, I. The FaCT system. In H. de Swart, editor, Automated Reasoning with Analytic Tableaux and Related Methods: *International Conference Tableaux'98, number 1397 in Lecture Notes in Artificial Intelligence*, pages 307--312. SpringerVerlag, Berlin, May 1998.
18. Richardson, R. and Smeaton, A. Using WordNet in a Knowledge-Based Approach to Information Retrieval. Technical Report CA-0395, *Dublin City Univ., School of Computer Applications*, Dublin, Ireland, 1995.
19. Rodriguez, A., Egenhofer, M. Determining Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering, Vol. 15*(2), pp. 442-456, March/April 2003.
20. Rodriguez, A., Egenhofer, M. Putting Similarity Assessments into Context: Matching Functions with the User's Intended Operations. *Context 99, Lecture Notes in Computer Science, Springer-Verlag*, pp. 310-323, September 1999.
21. Smith, M.K.,Welty, C., McGuinness, D.L. OWL Web Ontology Language Guide. W3C, http://www.w3.org/TR/2004/REC-owl-guide-20040210/. 10 February 2004.
22. Tversky, A. Features of Similarity. *Psychological Rev.*, Vol.84, pp. 327-352, 1977.
23. Visser, P., Jones, D., Bench-Capon, T., Shave, M. An Analysis of Ontology Mismatches; Heterogeneity versus Interoperability. *AAAI 1997 Spring Symposium on Ontological Engineering*, Stanford University, USA; also appeared as AAAI Technical Report SS-97-06, 1997.
24. 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, In *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, Seattle, WA, pp. 108-117, 2001.