Ontology Engineering Group
Laboratorio de Inteligencia Artificial
Dept. Inteligencia Artificial, Facultad de Informática – UPM

OEG Publication

Fernández-López M, Gómez-Pérez A, Ramos JA

OEGMerge: a case-based model for merging ontologies

MDAI 2006: Modeling Decisions for Artificial Intelligence
Edited by Torra V and Narukawa Y.
April 2006.
Tarragona, Spain.

Campus de Montegancedo, s/n – Boadilla del Monte – 28660 Madrid
Phone: +34 91 336 74 39 – Fax: +34 91 352 48 19

OEGMerge: a case-based model for merging ontologies

Mariano Fernández-López, Asunción Gómez-Pérez, José Ángel Ramos

OEG - Facultad de Informática. Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo, s/n. 28660 Boadilla del Monte. Madrid. Spain
{mfernandez@fi.upm.es, asun@fi.upm.es, jarg@fi.upm.es}

Abstract. No long ago ontology merging was a necessary activity, however, the current methods used in ontology merging present neither detailed cases nor an accurate formalization. For validating these methods, it is convenient to have a case list as complete as possible. In this paper we present the OEGMerge model, developed from the OEG (Ontological Engineering Group at UPM) experience, which describes precisely the merging casuistic and the actions to carry out in each case. In this first approach, the model covers only the taxonomy of concepts, attributes and relations.

1. Introduction

Ontologies can capture the agreed domain knowledge with a generic and formal form so that it can be reused and shared by applications and groups of people. From that definition, we can conclude erroneously that there is a unique ontology which models a particular domain. However, in literature we can find some ontologies that model the same knowledge domain adopting different criteria. There are, for example, several top-level ontologies which have different classification criteria for most of the general concepts in the taxonomy (i.e., [4]). In e-commerce, there are several standards and initiatives for service and product classification (UNSPSC[1], e-cl@ss[2], RosettaNet[3], etc.). This heterogeneity in ontologies also appears in domains such as medicine, law, art, sciences, etc.

Noy and Musen in [6] define: "*ontology alignment* consists in establishing different kinds of mappings (or links) between two ontologies, hence preserving the original ontologies; and *ontology merging* proposes to generate a unique ontology from the original ontologies". As we have seen, Noy and Musen´s ideas can also be applied to the merging of ontologies and thus, we have used them in this paper since it focuses on ontology merging. In fact, before merging two ontologies, we must establish some mappings between the ontology terms to merge.

Until now, the authors of the algorithms for merging simple ontologies did not provide the cases that can present when merging either ontologies or taxonomies.

---

[1] http://www.unspsc.org/
[2] http://www.eclass.de/
[3] http://www.rosettanet.org/

The Ontological Engineering Group (OEG[4]) can produce and implement the **OEGMerge** model for ontology merging, as a *plug-in* of WebODE [1]. This model specifies the cases found in different projects. One of the most relevant characteristic of this model is that it is not an algorithm but a rule-based system. In fact, the characteristics of the problems arisen when merging ontologies (use of incomplete knowledge, heuristic, ontological engineering knowledge, etc.) make us think that a rule-based model is more adecuated than a procedimental algorithm, being the ontologies and the intermapping set the knowledge base.

In section 2 of this paper, we analyze the merging methods proposed. In section 3, we show the method assumptions. In section 4, we introduce an example for case study. In section 5, we present the **OEGMerge** model and enumerate the cases and actions for merging. In section 6, we present a study of the completeness of the model cases. Finally, in section 7, we present the conclusions and future trends.

2. Ontology merging methods. State of the art.

The most important contributions to the merging of ontologies are ONIONS [3], FCA-Merge [8] and Prompt [7]. We shall explain them briefly.

ONIONS. With this method we can create a library of ontologies originated from different sources. In the first stages of the integration, the sources are unified without considering the generic theories that will be used in the modeling (part-whole or connectedness theories, for example). Later, these theories are incorporated as a base of the library and they are integrated as faces of the same polyhedron. In this way, the ontologies of the library are connected through the terms of the shared generic theories.

FCA-Merge [8]. This approach is very different from the approaches also presented in this section. FCA-Merge takes as input the two ontologies to be merged plus a set of documents on the domains of the ontologies, as shown in Figure 3.33. The merging is performed by extracting, from the documents, instances that belong to concepts of both ontologies. Thus, if the concept C_1 of the ontology O_1 has instances in the same documents as the concept C_2 of the ontology O_2, then C_1 and C_2 are candidates to be considered the same concept. To establish this relation between concepts and documents, we have created a table for each ontology. Each table relates each concept C of the associated ontology to the documents where instances of C appear. A lattice structure is generated from the tables and, finally, the merged taxonomy is obtained from the structure. At present, this method works only for lightweight ontologies.

The *PROMPT method* [7]. The main assumption of PROMPT is that the ontologies to be merged are formalized with a common knowledge model based on frames. A plug-in of Protégé-2000 merges ontologies according to this method. It proposes first to elaborate a list with the operations to be performed when merging the two ontologies (e.g., merging two classes, two slots, etc.). This activity is carried out automatically by the PROMPT plug-in. Then, a cyclic process starts. In each cycle the ontologist selects an operation from the list and executes it. Then a list of conflicts resulting from the execution of the operation must be generated, and the list of

[4] http://www.oeg-upm.net

possible operations for the following iterations is updated. Some of the new operations are included in the list because they are useful to solve the conflicts.

The mentioned methods represent the start of the ontology merging research. However, their authors do not provide details about the merging cases. In this paper, we present OEGMerge, a rule-based method which provides actions for each identified case. The rules have formalized *case* as antecedent and a*ction* as consequent. To do this we identify first the cases of the merging elements of the ontologies.

There are other systems for merging ontologies and schemas: OntoMerge[5], CUPID [5], GLUE [2], etc.

3. Assumptions

The assumptions of our method are:

1. It starts with two ontologies (Ontology 1 and Ontology 2) and a set of mappings between their components. This set has been able to be generated by hand or automatically.
2. The ontologies are correct.
3. The mappings are correct.
4. There is no contradiction between the two ontologies.
5. The two ontologies are represented following the frame paradigm.
6. Relations have not attributes. To represent relations with attributes, we reify them, that is, we use a concept instead of a relation.
7. The format of the mappings is:

   <term> <type of mapping> <term>

   where *term* can be a concept, an attribute, or a relation, and the type of mapping can be *similar*, *semantically similar*, or *subclass of*. The mapping *similar* is applied to components of the same type (a concept with a concept, an attribute with an attribute, etc.), and the mapping *semantically similar* is applied to different components (e.g. an attribute that matches with a relation in the other ontology). The mapping *subclass of* is applied to concepts. In the next section we present examples of mappings.
8. The cardinality of each mapping is 1:1. That is, a term of an ontology is mapped, at most, with a term of the other ontology.
9. Every mapping is exact, that is, the method does not manipulate similarity levels.
10. The considered components are: concepts, attributes, and relations. Therefore, instances and axioms are not considered during the merging process.
11. We assume closed world assumption hypothesis. Thus, when in our domain knowledge (ontologies and mappings) a relation between two components does not exist, we assume that these components are not related.

[5] http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html

Although it could be seen that the assumption presented here could not meet real world needs, there are activities identified in the ontology development process for checking ontology consistency or for analysing contradictions between two ontologies.

4. Example

Independently of the completeness of the domain knowledge, the cases presented in this paper cover all assumptions about the mappings, as shown in section 6.

Figure 1 presents Ontology 1. This ontology has 16 concepts, 12 taxonomic relations (*subclass of*) and 5 ad hoc relations.

Figure 2 presents Ontology 2. This ontology has 15 concepts, 10 taxonomic relations (*subclass of*) and 6 ad hoc relations.

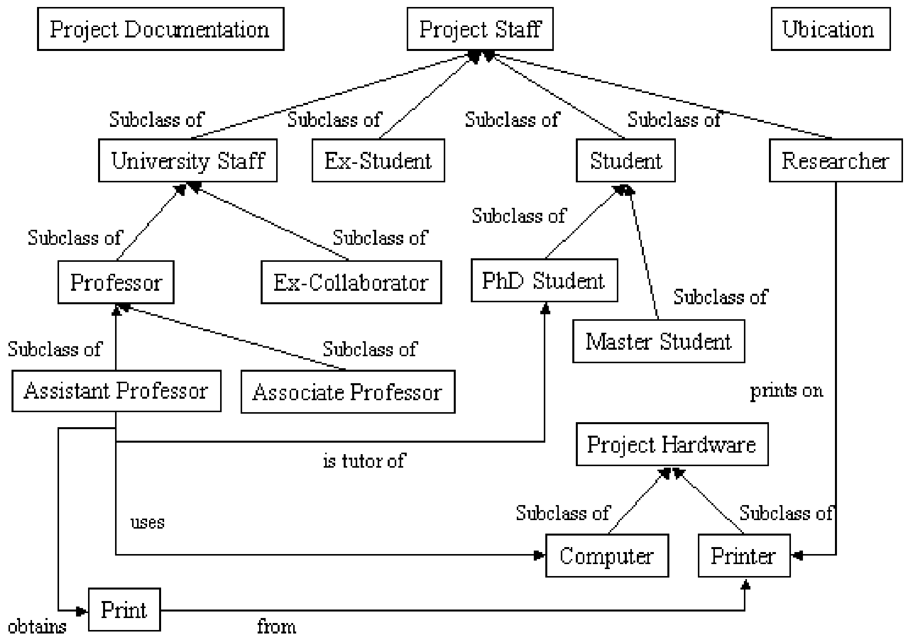

Figure 1: Ontology 1

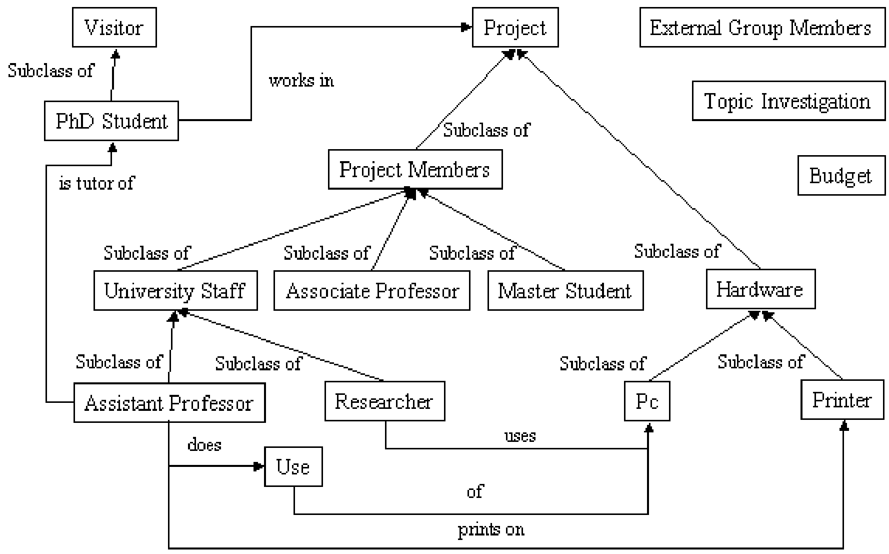

Figure 2: Ontology 2

Researcher [Ontology 1]	Researcher [Ontology 2]
String: Name	String: Name
String: Identify number	String: Last Name 1
String: E-mail	String: Last Name 2
String: Street/Avenue	String: Identify number
String: City	URL: Homepage
String: Country	String: Address
Computer: Associate computer	Printer: Associate printer
String: Topic Investigation	String: Ubication

Figure 3: Attributes of `Researcher`

Figure 3 presents the attributes of `Researcher` in each ontology.

The mappings between ontology components are detailed in Figure 4.

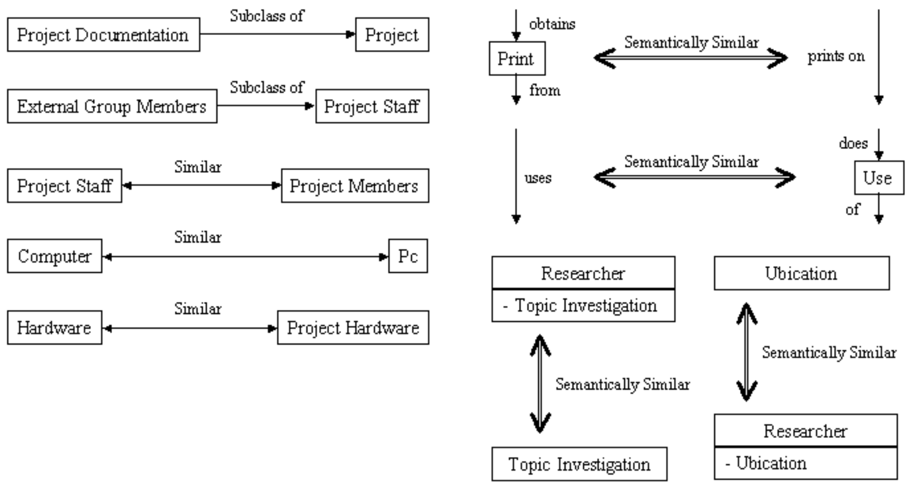

Figure 4: Set of mappings

To establish a point of reference to compare, we take Ontology 1 as a base ontology and Ontology 2 as an auxiliary ontology. This decision is not crucial since the final result is the same. However, experience tells us that for the present cases it is methodically better to choose the biggest ontology as base. The result ontology will be a modification of the base ontology, where the actions act.

5. Merging cases

Next, we are going to enumerate the merging cases grouped for a methodical review, using the previous example for better understanding.

Case 1: An auxiliary concept has not a similar base concept and a base concept, which is subclass of the auxiliary concept, exists.

In our ontology exemple, the concept `Project Documentation` (Ontology 1) is subclass of `Project` (Ontology 2), according to mappings.

In this case merging consists in including the auxiliary concept (`Project`) and the relation of mappings (`Project Documentation` *subclass of* `Project`) into the base ontology.

Case 2: A base concept has not a similar auxiliary concept, though an auxiliary concept, which is subclass of the base concept, exists

In our ontology example, the concept `External Group Members` (Ontology 2) is subclass of `Project Staff` (Ontology 1), according to mappings.

In this case merging consists in including the auxiliary concept (`External Group Members`) and the relation of mappings (`External Group Members` *subclass of* `Project Staff`) into the base ontology.

Case 3: An auxiliary concept has not a similar base concept, nor a subclass of a base concept, nor a superclass of a base concept.

In our ontology example, the concept `Budget` (Ontology 2) has not mapping with any base concept.

In this case merging consists in including the auxiliary concept `Budgetinto` into the base ontology.

Case 4: A base concept has not a similar auxiliary concept, nor a subclass of an auxiliary concept and nor a superclass of an auxiliary concept

In our example of ontologies, the concept `Ubcation` (Ontology 1) has not mapping with any auxiliary concept.

Merging, in this case, does not involve any action.

Case 5: A base concept has a similar auxiliary concept

In our ontology example, there are many base concepts similar to auxiliary concepts. For example, `Project Staff` is similar to `Project Members` (according to mappings) and `Printer` is similar to `Printer` (syntacticly).

However, this case has not a unique solution. The solution depends on the relation of the ancestor's concepts, and affects their attributes and relations. To distinguish these particular cases, we are going to enumerate them by adding the name case an "S" (from Similarity concepts).

Case S-1: A base attribute is similar to an auxiliary attribute

In our ontology example, in the base and auxiliary concepts `Researcher`, there exist a base attribute `Identify number` and an auxiliary attribute `Identify number`.

Evidently, the merging of this case does not involve any action.

Case S-2: A base attribute is not similar to any auxiliary attribute

In our example of ontologies, there exists a base attribute named `E-mail` which does not correspond with any auxiliary attribute of the auxiliary concept `Researcher` in the base concept `Researcher`.

Evidently, in this case merging does not involve any action.

Case S-3: An auxiliary attribute is not similar to any base attribute

In our ontology example, there exists a base attribute named `Homepage` which does not correspond with any base attribute of the base concept `Researcher` in the auxiliary concept `Researcher`

Merging, in this case, consists in including the auxiliary attribute `Homepage` in the base concept.

Case S-4: A base ad hoc relation is similar to an auxiliary ad hoc relation

In our ontology example, there exists a base ad hoc relation called `is tutor of` and an auxiliary ad hoc relation called `is tutor of` between the concepts `Assistant Professor` and `PhD Student,`

Evidently, in this case merging does not involve any action.

Case S-5: A base ad hoc relation is not similar to any auxiliary ad hoc relation
In our ontology example, there exists a base ad hoc relation between the concepts `Researcher` and `Printer` called `prints on`, but it does not exist in the auxiliary ontology.

Merging, in this case, does not involve any action.

Case S-6: An auxiliary ad hoc relation is not similar to any base ad hoc relation
In our ontology example, there exists an auxiliary ad hoc relation between the concepts `Researcher` and `Pc` called `uses`, but it does not exist in the base ontology.

Merging, in this case, consists in including the auxiliary ad hoc relation into the base ontology

Case S-7: An auxiliary ad hoc relation is semantically similar to a base concept and two base ad hoc relations
In our ontology example, there exist an auxiliary ad hoc relation called `prints on` between the auxiliary concepts (`Assitant Professor` and `Printer`), and a base concept called `Print` and her two ad hoc relations (`obtains` and `from`), being the origin of `obtains` the concept `Assistant Professor` and the destination of `from` the concept `Printer`. These two different conceptualizations are semantically similar, according the mappings. We are going to consider that, according to the assumptions presented in section 3, a concept has attributes and a relation has not.

Merging, in this case, does not involve any action.

Case S-8: A base ad hoc relation is semantically similar to an auxiliary concept and two auxiliary ad hoc relations
In our ontology example, there exist a base ad hoc relation called `uses` between base concepts (`Assitant Professor` and `Computer`) and an auxiliary concept called `Use` and its two ad hoc relations (`do` and `of`), being the origin of `do` the concept `Assistant Professor` and the destination of `of` the concept Pc (similar to `Computer`). These two different conceptualizations are semantically similar, according the mappings. We are going to consider that, according to the assumptions presented in section 3, a concept has attributes and a relation has not.

Here merging consists in including the auxiliary concept and the auxiliary ad hoc relations into the base ontology and eliminating the base ad hoc relation.

Case S-9: An auxiliary attribute is semantically similar to a base concept
In our ontology example, there exist an auxiliary attribute called `Ubication` in an auxiliary concept `Researcher` and a base concept called `Ubication`. These two different conceptualizations are semantically similar, according the mappings. We consider that the concept is a richer representation than the attribute.

Merging, in this case, consists in including an ad hoc relation called `has` between the concepts `Researcher` and `Ubication` in the base the ontology.

Case S-10: A base attribute is semantically similar to an auxiliary concept

In our ontology example, there exist a base attribute called `Topic Investigation` in base concept `Researcher` and an auxiliary concept called `Topic Investigation`. These two different conceptualizations are semantically similar, according the mappings. We consider that the concept is a richer representation than the attribute.

Merging here consists in including the concept `Topic Investigation` and an ad hoc relation called `has` between the concepts `Researcher` and `Topic Investigation` the in base ontology.

Case S-11: An auxiliary ad hoc relation is semantically similar to a base attribute

In our ontology example, there exist an auxiliary ad hoc relation called `uses` between the auxiliary concepts `Researcher` and `Pc`, and an attribute of the base concept `Researcher` called `Associate computer` whose type is `Computer`. These two different conceptualizations are semantically similar, according the mappings. We will consider that conceptualization of Ontology 2 is more intuitive than the conceptualization of Ontology 1.

Here merging consists in including the ad hoc relation `uses` between the concepts `Researcher` and `Computer` in the base ontology and eliminating the attribute `Associate computer`.

Case S-12: A base ad hoc relation is semantically similar to an auxiliary attribute

In our ontology example, there exist a base ad hoc relation called `prints on` between the base concepts `Researcher` and `Printer`, and an attribute of auxiliary concept `Researcher` called `Associate printer` whose type is `Printer`. These two different conceptualizations are semantically similar, according the mappings. We will consider that the conceptualization of Ontology 1 is more intuitive than the conceptualization of Ontology 2.

Merging here does not involve any action.

Case S-13: A base ancestor is similar to an auxiliary ancestor

In our ontology example, the base concept `Project Staff` (`University Staff`'s ancestor) is similar to the auxiliary concept `Project Members` (`University Staff`'s ancestor), according to mappings.

Merging here does not involve any action.

Case S-14: All base ancestors are no similar to any auxiliary ancestor

In our ontology example, the base concept `PhD Student` is similar to the auxiliary `PhD Student` (syntacticly), but the base ancestors (`Student`, `Project Staff`) are no similar to any auxiliary ancestors (`Visitor`).

Merging here consists in including the *subclass of* relation between the first auxiliar ancestor (e.g. `Visitor`) and the concept (`PhD Student`) in the base ontology.

Case S-15: the first auxiliary ancestor is similar to any brother base concept

In our ontology example, the auxiliary concept `Researcher` is *subclass of* the auxiliary concept `University Staff`, and the base concept `Researcher` is the brother (taxonomically) of the base concept `University Staff`.

Merging here consists in including the auxiliary *subclass of* relation (between `Researcher` and `University Staff`) in the base ontology and eliminating the base subclass of relation which is now redundant (between `Researcher` and `Project Staff`).

Case S-16: the first base ancestor is similar to any brother auxiliary concept
In our ontology example, the first ancestor of the base concept `Associate Professor` is the base concept `University Staff` and the auxiliary concept `Associate Professor` is the brother of the auxiliary concept `University Staff`.

Here merging does not involve any action.

6. Completeness of the model

Since the cases have been presented with examples, we are going to prove their completeness considering the dimensions of the type of component (concept, attribute and relation) and the type of mapping (similar [and semantically similar] and subclass of).

To do this, we can observe in Table 1 the relation of similarity, where the case which covers this situation appears.

Table 1: Cases between similar elements

Similarity	Base concept	Base attribute	Base ad hoc relation
Auxiliary concept	**5**	**S-10**	**S-8**
Auxiliary attribute	**S-9**	**S-1**	**S-12**
Auxiliary ad hoc relation	**S-7**	**S-11**	**S-4**

For example, as Table 1 shows, the case of an auxiliary concept being similar to a base concept is case 5; the case of an auxiliary ad hoc relation being similar to a base attribute is case S-11; the case of an auxiliary attribute being similar to a base ad hoc relation is case S-12.

For these elements that are not similar to any other, consult Table 2.

Table 2: Cases without similarity

	Don't exist similar
Base concept	**4**
Auxiliar concept	**3**
Base attribute	**S-2**
Auxiliary attribute	**S-3**
Base ad hoc relation	**S-5**
Auxiliary ad hoc relation	**S-6**

So, for example, the case of a base concept no similar to another auxiliary element is case 4; the case of an auxiliary attribute no similar to another base element is case S-3.

For the cases of taxonomic differences, the study of subclass of relation, we present tables 3, 4 and 5 with three pairs of comparisons: *subclass of* between concepts, transitive *subclass of* with ancestors of similar concepts, and *subclass of* between parents and brothers of similar concepts. The cases of other combinations of similarity between concepts of the two ontologies (the auxiliary concept 1 is not similar to the base concept 1) are already considered in the case 5.

Table 3: Cases of simple mappings through *subclass of*

	Mapping = subclass of
Base concept mapped to auxiliary concept	**1**
Auxiliary concept mapped to base concept	**2**

Table 4: Cases of mappings through *subclass of* when the auxiliary concept 1 is similar to the base concept 1 and the auxiliary concept 2 is similar to the base concept 2

	Base concept 1 is subclass of auxiliary concept 2	Base concept 1 is brother of auxiliary concept 2
Base concept 1 is subclass of auxiliary concept 2	**S-11**	**S-15**
Base concept 1 is subclass of auxiliary concept 2	**S-16**	**Already considered in 5**

Table 5: Cases of mappings through *subclass of* when the auxiliary concept 1 is similar to the base concept 1 and the auxiliary concept 2 is NOT similar to the base concept 2

	Base concept 1 is subclass of auxiliary concept 2	Base concept 1 is brother of auxiliary concept 2
Base concept 1 is subclass of auxiliary concept 2	**S-14**	**Already considered in 5**
Base concept 1 is subclass of auxiliary concept 2	**Already considered in 5**	**Already considered in 5**

7. Conclusions and future trends

The Ontological Engineering Group has established a casuistic for merging ontologies, which is presented in this paper with the help of the OEGMerge model. This casuistic is the result of our experience in the field of ontology merging. In this work we propose the actions required to carry out the merging of cases of concepts, attributes and relations (three of the most basic components of any ontology under the frame paradigm).

We, the Ontological Engineering group of UPM, have tested the model here presented in very many merging processes.

It remains for the future to amplify ODEMerge so as to cover those components that are not currently covered (axioms, references, groups, etc.,) under the frame paradigm and other paradigms. As it has already been stated, it would be necessary to consider, in the mapping definition, new predicates to express correspondences with more semantics.

9. Acknowledgements

This work has been partially supported (the formalization of the model) by the National Project "Infraestructura tecnológica de servicios semánticos para la web semántica" (TIN-2004-02660). We are also grateful to Rosario Plaza for her help in checking and reviewing the grammar and spelling of the paper and improving clarity and style.

10. References

1. Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A (2003) *WebODE in a nutshell*. AI Magazine, 24(3)-37-47
2. Doan A, Madhavan J, Domingos P, Halvey A (2002) *Learning to map between ontologies on the semantic web*. In Proceedings of 11th International WWW Conference, Hawaii (2002)
3. Gangemi A, Pisanelli DM, Steve G (1999) An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. Data & Knowledge Engineering 31(2):183–220
4. Guarino N, Welty C (2000) A Formal Ontology of Properties. In: Dieng R, Corby O (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 97–112
5. Madhavan, J., Bern-stein, P. A., Rahm, E.: Generic schema matching with Cupid. VLDB Journal (2001) 49–58
6. Noy NF, Musen MA (1999) *SMART: Automated Support for Ontology Merging and Alignment* In: Gaines BR, Kremer B, Musen MA (eds) 12th Banff Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Alberta, Canada, 4-7:1–20
7. Noy NF, Musen MA (2000) *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment* Proceedings of the Seventeenth National Conference on Artifical Intelligence (AAAI-2000). Austin, Texas. 2000.
8. Stumme G, Maedche A (2001) *FCA-Merge: Bottom-Up Merging of Ontologies* IJCAI 2001 – Proceedings of the 17th International Joint Conference on Artificial Intelligent. Seattle, USA, Agosto 2001.