Semantic Web Complex Ontology Mapping

Nuno Silva and João Rocha
GECAD - Knowledge Engineering and Decision Support Research Group
Instituto Superior de Engenharia do Porto
4200-072 Porto – Portugal
Nuno.Silva@dei.isep.ipp.pt; jrocha@ipp.pt

Abstract

Ontology mapping is the process whereby two ontologies are semantically related at conceptual level and the source ontology instances are transformed into target ontology entities according to those semantic relations. Ontology mapping faces new challenges in the context of Semantic Web, especially concerning heterogeneity, dynamics, distribution and limitations on representation technology. This paper introduces a new methodology and transformation process based on the notion of Service, which represents system transformation capabilities. MAFRA Toolkit is a specific implementation of MAFRA-Mapping FRAmework, where these new methodology and transformation process are being validated. MAFRA Toolkit is being applied in the European project Harmonise, which aims to provide solutions for (semi-) automatic interoperability between major operators in tourism e-business. MAFRA plays a major role in the specification, representation and reconciliation phases of the semantic mapping within the scope of the Harmonise technology.

1. Introduction

Semantic Web, the next WWW trend, suggests the annotation of Web resources with machine-processable metadata, which can provide tools to analyse meaning and semantic relations between documents and their parts. Ontologies as means for conceptualizing and structuring knowledge are seen as the key to the realization of the Semantic Web vision. Ontology allows the explicit specification of a domain of discourse, which permits to access to and reason about an agent knowledge. Ontologies raise the level of specification of knowledge, incorporating semantics into the data, and promote its exchange in an explicit understandable form. Semantic Web and ontologies are therefore fully geared as a valuable framework for distinct applications, namely business applications like E-Commerce and B2B. However, ontologies do not overcome *per se* any interoperability problems, since it is hardly conceivable that one ontology is applied for all kind of domains and applications. Ontology mapping does not intend to unify ontologies and their data, but to transform ontology instances according to the semantic relations (mapping relations) defined at conceptual level. Repositories are therefore kept separated, independent and distinct, maintaining their complete semantics and contents. This new proposed approach adopts a declarative specification of mappings, hiding the procedural complexity of specification and execution, while its preconized open architecture allows the integration of new mapping relations into the system, improving mapping capabilities.

MAFRA current approach is being used and tested under the Harmonise project (http://www.harmonise.org). Harmonise intends to overcome the interoperability problems occurring between tourism operators due to use of different information representation standards. MAFRA Toolkit was adopted as the representation and transformation engine core technology for the Harmonise project. Harmonise uses an "Interoperability Minimum Harmonisation Ontology" (IMHO) as *lingua franca* between agents. MAFRA Toolkit is responsible for the acquisition, representation and execution of the ontology mapping between each agent specific ontology and IMHO. Harmonise examples will further illustrate MAFRA capabilities during this paper.

Relevant projects in the area of ontology mapping will be described and compared, in section 2, while section 3 presents essential backgrounds of this work. The new ontology mapping methodology is introduced and exemplified in section 4. In section 5, the newly developed transformation process is described in detail. The work presented in core sections 4 and 5 represents new contributions for the ontology mapping research field. Finally, Section 6 will provide an overview of the achieved results and point out current and future efforts.

2. Related work

Much valuable work has been done in the area of distributed databases in integrating and exchanging data between entities, but the work described here focus on the Semantic Web environment, its characteristics, constraints and technologies.

Both the centralized mediated data schema and ontology merging adopt a centralized approach in which data source schemas/ontologies are unified into a single schema/ontology. The "centralized" approach of the mediation is probably not flexible enough to be scaled up to the Web. Consequently, ontology mapping is empirically perceived as a more dynamic knowledge sharing process than centralized schema mediation or ontology merging, and arises naturally as a potential solution for ontology-based interoperability problems. Three distinct ontology mapping projects are considered paradigmatic approaches. In [1] authors describe an extension to Protégé to map between domain ontologies and problem solving methods. This approach defines a valuable set of desiderata and mapping dimensions, but its known implementation lacks some important features especially in allowing mapping between multiple concepts. Yet, there is no record of experiments that apply it to the Semantic Web environment. The second approach is RDFT [2], a meta-ontology that describes Equivalence and Versioning relations between either an XML DTD or RDFS document and another XML DTD or RDFS document. An RDFT instantiation describes the semantic relations between source and target documents, which will be further applied in the transformation of documents. Thirdly, the Buster project [3] applies information integration to the GIS domain. Two distinct approaches were proposed: rule-based transformation and re-classification. The rule-based approach applies a procedural transformation to instance properties, while classification applies class membership conditions to infer target classification through description-logic tools. However, these two approaches are not integrated, which limits mapping capabilities.

3. MAFRA and SBO background

MAFRA-MApping FRAmework [4]. is the known ontology mapping framework that integrates the largest number and the most encompassing phases of the ontology mapping process. Semantic Bridge Ontology (SBO) [4] is the MAFRA companion respecting description and representation of ontology mappings. In the scope of this paper, ontologies are knowledge models represented in RDFS.

The core concept in SBO is the semantic bridge, which is a declarative representation of a semantic relation between source and target ontologies entities. Semantic bridges provide the execution engine with the necessary information to transform instances of certain source entities into instances of certain target entities.

One of the innovations in MAFRA Toolkit corresponds to its service-centric approach. Each semantic bridge has an associated transformation service that determines the transformation procedure and the information the user must provide to the transformation engine. Each Service is characterized by a set of arguments, which in turn are characterized by name, type, optionality and location (whether it is a source, target or condition argument). Services are not only responsible for the transformation capabilities but also for the validation of argument values and semi-automatic mapping. Deeper details on service-oriented approach can be found in [5]. Table 1 presents some examples of Services currently available in MAFRA Toolkit.

Table 1. Some MAFRA prototypal mapping relations and their interface

CopyInstance	**Creates target concept instances for each source concept instance that fulfils conditions.**			
Argument ID	**Type**	**O**	**L**	**Comment**
Source Concept	Class	N	S	Source ontology class whose instances will be transformed
Extensional Specification	ArrayOfConditions	Y	S	Extensional definition of source class instances
Target Concept	Class	N	T	Target ontology class to create
Generic Conditions	ArrayOfConditions	Y	C	Constraint of the bridge execution
Minimum Cardinality	Integer	Y	C	The minimum number of instances to translate using each instance
Maximum Cardinality	Integer	Y	C	The maximum number of instances to translate using each instance
CopyRelation	**Creates a relation between concepts instances based**			
Source Path	Path	N	S	Source ontology path for each path the bridge will be executed
Extensional Specification	ArrayOfConditions	Y	S	Extensional definition of source class instance
Target Path	RelationPath	N	T	Target ontology path to create
Generic Conditions	ArrayOfConditions	Y	C	Conditions based on source entities values
Maintain Context	Boolean	Y	C	Only the extensional specified context is used
Minimum Cardinality	Integer	Y	C	The minimum number of instances to translate using each instance
CopyAttribute	**Copies (no changes) the source property value (string) to the target property.**			
Source Attribute	AttributePath	N	S	Source ontology attribute whose instances will be copied

Target Attribute	AttibutePath	N	T	Target ontology attribute to copy to
Generic Conditions	ArrayOfConditions	Y	C	Conditions based on source entities values
Maintain Context	Boolean	Y	C	Only the extensional specified context is used
Minimum Cardinality	Integer	Y	C	The maximum number of instances to translate using each instance
Maximum Cardinality	Integer	Y	C	The maximum number of instances to translate using each instance
CountRelations	**Counts the number of instances of a relation.**			
Source Path	Path	N	S	Source ontology path to copy
Target Attribute	AttributePath	N	T	Target ontology attribute to create
Extensional Specification	ArrayOfConditions	Y	C	Extensionally defines source class instance
Generic Conditions	ArrayOfConditions	Y	C	Conditions based on source entities values
Maintain Context	Boolean	Y	C	Only the extensional specified context is used
Split	**Splits by separators, the literal in source attribute.**			
Source Attribute	AttributePath	N	S	Source ontology attribute whose instances will be splited
Separators	ArrayOfLiterals	N	S	Literals or regular expressions to split by
Target Attributes	ArrayOfAttributePaths	Y	T	List of attributes to instantiate with splited values
Generic Conditions	ArrayOfConditions	Y	C	Conditions based on source entities values
Maintain Context	Boolean	Y	C	Only the extensional specified context is used

4. MAFRA mapping methodology

The MAFRA methodology provides rules and guidelines in defining semantic relations using SBO in coordination with the execution process introduced in section 5. Both sections are mutually complementary.

In next sections examples will be used. The presented examples try to use wide acceptable semantic relations, but typical semantic relations tend to be subjective.

4.1. Object-oriented approach

SBO model suggests a hierarchical, object-oriented approach of semantic bridges definition. Accordingly, the MAFRA methodology advises its use when possible, promoting modularity, reusability and readability of the mapping definition. Consider Figure 1 where excerpts of two ontologies are represented in UML.

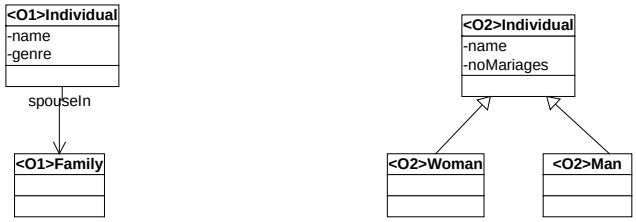


Figure 1. Excerpts of two ontologies

Both source and target ontologies define the concept Individual, which according to domain expert should be mapped. Because the target entity is a concept, the CopyInstance service will be used (concept bridge):

```
Individual2Individual=semanticBridge(
   CopyInstance,
   {SourceConcept=<O1>Individual},
   {TargetConcept=<O2>Individual},
   {abstract=True} )
```

<O2>Individual represents an abstract concept since it is never instantiate. This means that there are no <O2>Individual instances, but only <O2>Man and <O2>Woman instances. Accordingly, the Individual2Individual bridge is state abstract. In fact, <O1>Individual will originate either <O2>Woman or <O2>Man, depending on the value of the <O1>Individual.genre attribute in each instance. Consequently, two concept bridges are to be defined, each requiring a condition expression to check the value of <O1>Individual.genre. If equals "M", a <O2>Man instance will be created, otherwise a <O2>Woman instance will be created.

```
Individual2Woman=semanticBridge(
   CopyInstance,
   {SourceConcept=<O1>Individual},
   {TargetConcept = <O2>Woman>},
   {<O1>Individual.genre="F"} )
Individual2Man=semanticBridge(
   CopyInstance,
   {SourceConcept=<O1>Individual},
   {TargetConcept=<O2>Man>},
   {<O1>Individual.genre="M"} )
subBridgeOf( Individual2Woman,
             Individual2Individual )
subBridgeOf( Individual2Man,
             Individual2Individual )
```

Several comparison operators are available in the current implementation of MAFRA, namely Equal, Different, Match (regular expression match) and Like (*a la* SQL substring match). Or, And, Xor and Not logic operators are also available, allowing the specification of complex conditional expressions. For example, one could specify a more elaborated condition for Individual2Man bridge:

```
{  <O1>Individual.genre="M" Or
   <O1>Individual.genre Like "Male" Or
   <O1>Individual.genre Match "^M*" }
```

4.2. Alternative bridges

The specification of conditions in independent bridges might not be sufficient for controlling the execution of the mapping. In fact, definition of conditions ensures that the bridge is executed only if the condition holds, but it does not ensure that other bridges are not executed. This is not a problem in the scenario of Figure 1 since an <O1>Individual instance is either man or woman. However, this situation can occur in case no sufficient conditions are specified for a bridge. In order to explicitly express this constraint, the alternativeBridges construct might be applied:

```
alternativeBridges(
   list( Individual2Woman,Individual2Man) )
```

4.3. Property Bridges

Once concept bridges are defined, property bridges are defined and attached to concept bridges. In the running example, two property bridges are to be defined. Bridge name2name relates <O1>Individual.name to <O2>Individual.name, and bridge spouseIn2noMariages relates <O1>Individual.spouseIn to <O2>Individual.noMariages. Property bridge name2name will use the CopyAttribute service, since no transformation in the source values is required. On the other hand, the spouseIn2noMariages property bridge will use the CountRelation service.

```
name2name=semanticBridge(
   CopyAttribute,
   {SourcePath=<O1>Individual.name},
   {TargetPath=<O2>Individual.name},
   {} )
spouseIn2noMariages=semanticBridge(
   CountRelation,
   {SourcePath=<O1>Individual.spouseIn},
   {TargetPath=<O2>Individual.noMariages},
   {} )
hasBridge( Individual2Individual,
   name2name )
hasBridge( Individual2Individual,
   spouseIn2noMariages )
```

Remark that property bridges are defined within the scope of the Individual2Individual concept bridge, but they will be executed within the scope of Individual2Man and Individual2Woman, since these are sub-bridges of Individual2Individual, and this is an abstract concept bridge. This hierarchical inheritance mechanism, similar to object-oriented modelling, profits and provides the benefices recognized to object-oriented approach, specially when applied to distributed, well-structured, inter-dependent ontologies, which are progressively adopted and supported in Semantic Web [6].

5. Execution process

The execution process corresponds to the MAFRA execution module, where source ontology instances are actually transformed/translated into target ontology instances. The execution process comprises two phases:

- In first phase all concept bridges run for each and all instances of the source concept defined in concept bridge. This phase creates the target instances and attaches them a new identity;
- In second phase property bridges are executed in scope of each newly created instance. The property bridges to execute are those defined in scope of concept bridge and those defined in scope of the super concept bridge, if some exists.

This clear separation between the execution of concept bridges and property bridges allows increased modularization and simplicity of transformation. The Copy Relation service is paradigmatic of this modularization and simplicity. Imagine linking the instance to another one that do not yet exists. In such situation, it would be necessary to run the concept bridge responsible for the creation of the target entity in the scope of the first concept bridge. Furthermore, all property bridge of second concept bridge would be executed to, which could again imply the execution of concept bridges, and so on.

Instance transformations are performed by the CopyInstance service, which is part of MAFRA core system and cannot be substituted by another externally, provided service. This constraint is necessary due to specificities of instance transformation requirements. In special, these specificities have important impact in CopyRelation service. In fact, these are the fundamental services required to manage concept instances (i) CopyInstance create the instances and (ii) CopyRelation inter-relates them.

5.1. Instance transformation

Due to the need to copy source instances relations to target instances, it is necessary to maintain information about which source instance gave rise to which target instance. The CopyInstance service is responsible for this task. The so-called Transformation Table is used to maintain this information, through a list of tuples in the form of:

```
< source_instance_id, extensional_specification,
   target_instance_id, concept_bridge_id >
```

A reference to the concept bridge responsible for the creation of the target instance is kept in the table in order to access its property bridges (that are defined in the scope of the concept bridge) in second phase of the execution

process. Extensional specification is responsible for the univocal specification of source instance, necessary when many target instances are created from the same source instance. It will be described below.

Consider the mapping scenario of Figure 2 respecting excerpts of TourInFrance (http://www.tourisme.gouv.fr) and SIGRT (http://www.dgturismo.pt/irt/c_pi.asp) ontologies. TourInFrance (namespace TIF) is the source ontology and SIGRT (namespace SIGRT) is the target ontology. SIGRT ontology is composed of three interrelated concepts, while TourInFrance ontology is composed of two inter-related concepts.

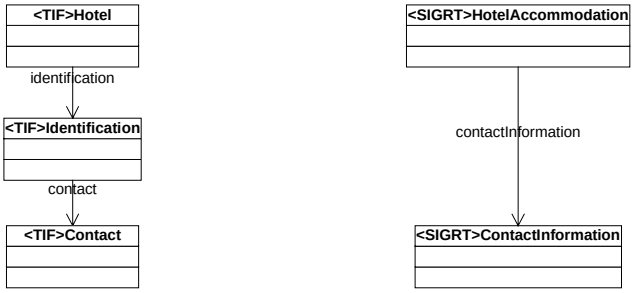


Figure 2. Excerpts of TourInFrance and SIGRT ontologies.

Two concept bridges are easily perceived (<TIF>Identification has no counterpart in SIGRT ontology and therefore is not mapped):

```
Hotel2HotelAccommodation=semanticBridge(
   CopyInstance,
   {SourceConcept=<TIF>Hotel},
   {TargetConcept=<SIGRT>HotelAccommodation},
   {} )
Contact2ContactInformation=semanticBridge(
   CopyInstance,
   {SourceConcept=<TIF>Contact},
   {TargetConcept=<SIGRT>ContactInformation},
   {} )
```

Figure 3 represents the target instances obtained from source ontology instance when executing previous concept bridges. The Transformation Table resulting from these semantic bridges execution can be found in Figure 5.

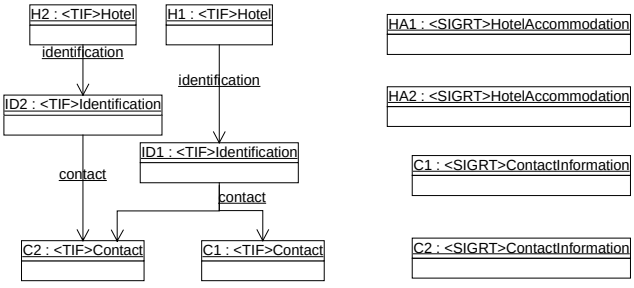


Figure 3. Source and target instances transformation

5.2. Extensional Specification

Source instance Extensional Specification is necessarily applied when many target instances are created from one single source instance and those target instances need to be related. This approach is based on Description Logics basic principles.

Extensional specification complements the instance identity with values of instance properties to create a virtual instance in the Transformation Table. Hence each pair of source identity and extensional specification is unique in the Transformation Table. Each of these pairs relate to only one target entity, providing the means to univocally choose between many target instances derived from the same source instance.

In order to illustrate this approach, consider the inverse scenario of Figure 2, where the SIGRT ontology is to be mapped to TourInFrance. Two ConceptBridges are easily perceived:

```
HotelAccommodation2Hotel=semanticBridge(
   CopyInstance,
   {SourceConcept=<SIGRT>HotelAccommodation},
   {TargetConcept=<TIF>Hotel},
   {} )
ContactInformation2Contact=semanticBridge(
   CopyInstance,
   {SourceConcept=<SIGRT>ContactInformation},
   {TargetConcept=<TIF>Contact,
   {} )
```

The fore mentioned problem arises because no source concept is directly related to the <TIF>Identification, and because it is fundamental to create instances of <TIF>Identification in order to recreate the relation between <SIGRT>HotelAccommodation and <SIGRT>ContactInformation. The general solution advises to semantically map the target concept with one or more property values of a concept. This imply the use of the extensional specification approach.

In this particular example, the domain expert decides that one <TIF>Identification instance should be created for each <SIGRT>HotelAccommodation/contactInformation property value. This means that for each <SIGRT>ContactInformation instance related to <SIGRT>HotelAccommodation, an instance of <TIF> Identification will be created. According to the CopyInstance service interface, the extensional specification is a source argument. Here is the concept bridge specification:

```
HotelAccommodation2Identification=semanticBridge(
   CopyInstance,
   {SourceConcept=<SIGRT>HotelAccommodation,
    foreach(
      <SIGRT>HotelAccommodation/contactInformation},
   {TargetConcept=<TIF>Hotel} )
```

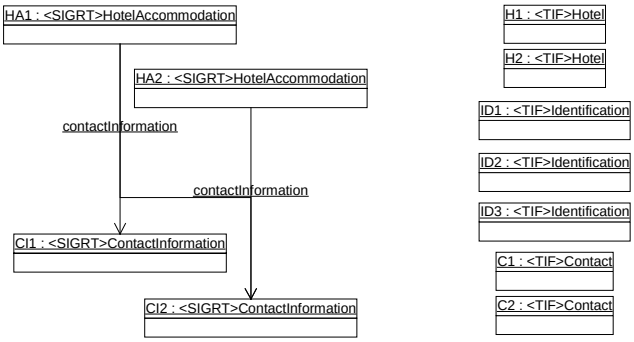


Figure 4. Source and target instance transformation representation

Figure 4 represents the set of source instances and the obtained set of target instances after the execution of the three previous concept bridges. Table 2 represents the resulting Transformation Table.

Table 2. Transformation Table declaring extensional specification of source instances

Source Instance ID	Source Instance Extensional Specification	Target Instance ID	Concept Bridge
HA1		H1	HA2H
HA2		H2	HA2H
HA1	contactInformation=CI1	ID1	HA2I
HA1	contactInformation=CI2	ID2	HA2I
HA2	contactInformation=CI2	ID3	HA2I
CI1		C1	CI2C
CI2		C2	CI2C

5.3. Inter-relation of instances

The creation of relations between instances is achieved through the CopyRelation service. Besides it is substitutable, it is expected that the provided implementation would fulfil all requirements associated with the copy relation process. In its simplest instantiation, CopyRelation service takes two arguments:

- The source path that will provide the value to iterate through in creating the target relation. The source path can be either an attribute or a relation path;
- The target relation path to instantiate.

This simple form is enough to complete the mapping scenario of Figure 2, relation HotelAccommodation to ContactInformation:

```
contact2contactInformation=semanticBridge(
   CopyRelation,
   {SourcePath=<TIF>Hotel/identification/contact},
   {TargetPath=<SIGRT>HotelAccommodation/
                              contactInformation},
   {} )
hasBridge( Hotel2HotelAccommodation,
           contact2contactInformation )
```

The first step in the execution of CopyRelation service is to project all paths values of the source and condition arguments. This step enumerates all possible combinations of values. Including the condition paths in the projection simplifies the verification of conditions and includes it in the same process.

CopyRelation tries to instantiate the target relation <SIGRT>HotelAccommodtion/contactInformation for each instance created for <SIGRT>HotelAccommodation.

Source Instance ID	Source Instance Extensional Specification	Target Instance ID	Concept Bridge
H1		HA1	H2HA
H2	2	HA2	H2HA
C1		CI1	C2CI
C2		CI2	C2CI

1 Projection Table	
identity	Contact
ID1	C1
ID1	C2

Figure 5 - Transformation and projection tables' example.

Figure 5 depicts the projection table created applying the contact2contactInformation semantic bridge in scope of instance HA1. In this case, two combinations are derived because source instance H1 is related to ID1, which is in turn related to two <TIF>Contact instances (C1 and C2). The source instance identity for each row of the projection table is then searched in the Transformation Table (step 1). The respective target instance (step 2) corresponds to the value to instantiate in the target relation. This process is executed for all lines of the projection table that fulfil the conditions associated, including cardinality of the created relation.

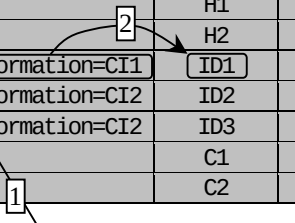
The CopyRelation service requires extra information if the target concept is created through extensional specification. There is the need to extensionally specify the source instance that gave rise to the target instance. Consider the inverse scenario of Figure 2 (instantiated in Figure 4). The following semantic bridges are needed:

```
contactInformation2identification=semanticBridge(
   CopyRelation,
   {SourcePath=<SIGRT>HotelAccommodation/
              contactInformation/ContactInformation,
    ExtensionalSpecification=foreach(
      <SIGRT>HotelAccommodation/contactInformation)
   },
   {TargetPath=<TIF>Hotel/identification},
   {} )
hasBridge( HotelAccommodation2Hotel,
   contactInformation2identification )
contactInformation2contact=semanticBridge(
   CopyRelation,
   {SourcePath=<SIGRT>HotelAccommodation/
```

```
                        contactInformation},
   {Target Path=<TIF>Hotel/identification},
   {} )
hasBridge( HotelAccommodation2Identification,
           contactInformation2contact )
```

Figure 6 illustrates the execution process for the creation of relation between H1 and ID1. The projection table includes all source properties specified as arguments: <SIGRT>HotelAccommodation as Source Path argument and <SIGRT>HotelAccommodation/contactInformation as extensional specification argument. HA1 source ontology instance is the source instance for both H1 and ID1. The extensional specification in the CopyRelation bridge, complements the identification of the target instance (step 1), allowing the correct identification of the target instance (step 2).

Source Instance ID	Source Instance Extensional Specification	Target Instance ID	Concept Bridge
HA1		H1	HA2H
HA2		H2	HA2H
HA1	contactInformation=CI1	ID1	HA2I
HA1	contactInformation=CI2	ID2	HA2I
HA2	contactInformation=CI2	ID3	HA2I
CI1		C1	CI2C
CI2		C2	CI2C



Projection Table	
identification	Extensional specification
HA1	contactInformation=CI1
HA1	contactInformation=CI2

Figure 6. Transformation and projection tables with extensional specification constraints

Several other arguments are available to control the execution of bridges, namely the cardinality and control of conditions, but their functionality is service dependent.

6. Conclusion

This paper puts forward a new methodology and transformation process to ontology mapping, combining rule-based transformation with Description Logic-like approaches. This combination originates a very powerful ontology mapping system. MAFRA standard transformation services would provide support for very complex ontology mapping problems, but its architecture allows for an easy integration of additional services, dealing with an important characteristic of ontologies on the semantic web: heterogeneity. The MAFRA Toolkit (publicly available at Source Forge) implements the ideas described in this paper, providing domain expert with an intuitive, easy to use and integrated GUI.

Experiences in Harmonise project show that MAFRA approach fulfils all requirements found during the mapping processes. Domain experts achieved to map very different ontologies, ranging from well structured to completely flat structures. They reported the need for the combination of services, instead of development of new ones. This feature will be provided in next releases, improving modularization. For the moment the Toolkit is very stable and efficient, but further formal tests are required.

Currently, the automatic creation of semantic bridges is being developed. Service-oriented approach is once again applied along. The automatic identification of semantic relations is achieved through the use of external knowledge sources (dictionaries, WordNet). Negotiation of ontologies between different partners and evolution of mappings according to ontology evolution are some of our next efforts.

7. Acknowledgements

This work is partially supported by the Portuguese MCT-FCT project POCTI/2001/GES/41830. Many thanks to Alexander Maedche for his ideas and support. Thanks to Oliver Fodor and Mirella Dell'Erba for their feedback on the application of MAFRA Toolkit in the Harmonise project. Thanks to Jorge Santos for our discussions.

8. References

[1] Park, J. Y., Gennari, J. H., and Musen, M. A. Mappings for Reuse in Knowledge-based Systems. Banff, Canada. 11th Workshop on Knowledge Acquisition, Modelling and Management (KAW 98). 1998.

[2] Omelayenko, B. RDFT: A Mapping Meta-Ontology for Business Integration. 76-83. Lyon, France. Proceedings of the Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at ECAI'2002. 2002.

[3] Stuckenschmidt, H. and Wache, H. Context Modeling and Transformation for Semantic Interoperability. 115-126. Knowledge Representation Meets Databases. 2000.

[4] Maedche, A., Motik, B., Silva, N., and Volz, R. MAFRA - An Ontology MApping FRAmework in the Context of the Semantic Web. 235-250. Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02). 2002.

[5] Silva, N. and Rocha, J. E-Business Interoperability Through Ontology Semantic Mapping. Lugano, Switzerland. PRO-VE'2003. 2003.

[6] Maedche, A., Motik, B., Stojanovic Ljiljana, Studer, R., and Volz, R. An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies. 439-448. Budapest, Hungary. Proceedings of the WWW 2003. 2003.