ONTOLOGY MAPPING FOR INTEROPERABILITY IN SEMANTIC WEB

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

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. The objective of MAFRA–MApping FRAmework – is to cover all the phases of the ontology mapping process, including analysis, specification, representation, execution and evolution. The MAFRA Toolkit is an implementation of MAFRA, adopting an open architecture in order to observe the Semantic Web requirements, namely performance and transformation capabilities. One of the MAFRA Toolkit novelties respects its service-oriented approach, which claims that the capabilities of an ontology mapping system dependent on what transformations are present. Independent, plug able services are then responsible for the instances transformations, but they also provide support for other ontology mapping tasks like automatic specification of semantic relations, negotiation and evolution.

KEYWORDS

Semantic Web, Systems Integration, Ontology, Ontology Mapping

1. INTRODUCTION

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 explicitly 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 a single ontology is applied in 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.

The work described in this paper has been developed in scope of MAFRA-MApping FRAmework (Maedche et al 2002). MAFRA covers all the phases of the ontology mapping process, including analysis, specification, representation, execution and evolution. Current MAFRA implementation is MAFRA Toolkit. It adopts a declarative specification of mappings, hiding the procedural complexity of specification and execution, while its open architecture allows the integration of new semantic relations into the system, improving mapping capabilities as required.

In Section 2, relevant background technology and projects in the area of knowledge interoperability will be analyzed and compared. Section 3 presents the MAFRA Toolkit service-oriented architecture. Section 4 describes the automatic semantic bridging process while section 5 describes and exemplifies the execution process. Section 6 describes the query web service, which allows independent agents to interoperate based on ontology mapping and Section 7 presents some experiences. Finally Section 8 provides an overview of the achieved results and point out some current and future efforts.

2. SEMANTIC WEB CONTEXT

Two distinct research fields assume particular relevance for the ontology mapping process: (i) the semantic web representation languages, and (ii) the projects related to information integration, especially ontology mapping projects.

Ontology Languages for the Semantic Web

Ontology representation languages for the Semantic Web are in early stages of development. This is the case of the OWL Web Ontology Language (W3C 2003), the eventual future W3C recommendation for ontology language. Instead, other ontology representation languages like DAML, OIL, DAML+OIL or even RDFS, are being used. All of them rely on the Resource Description Framework (RDF) subject-predicate-object model, which provides a basic but extensible and portable representation mechanism for the Semantic Web. RDFS is an RDF extension used to describe simple vocabularies (concepts and properties) and interrelations (e.g. taxonomies). DAML, OIL, DAML+OIL and OWL are RDFS extensions arising as powerful ontology representation languages. OWL will provide most of the DAML+OIL functionalities, and will adopt a layered structure introduced in OIL, distinguishing between different levels of requirements respecting ontological description. The highest the layer, the more powerful the language become, but also the more powerful processing capabilities become necessary.

The ontology mapping process highly depends on the representation language and the ontology semantic details, at both levels of the process:

- At instance level, instances are classified according to ontology. The more complex the definition of concepts is, more difficult the classification becomes. Because classification of instances is a distinct research field, we rely on third part tools for carry such task. Several ontology tools (e.g. Ontobroker (Ontoprise 2003), FaCT (Horrocks 2003)) support automatic and inferred classification of individuals.
- At ontological level, entities from two ontologies are semantically related. The more detailed and extensive the ontology is, the more information is available for analysis and comparison, leading to potentially better semantic relations. The ontology mapping process must exploit the ontology contents to better derive semantic relations, with minimum human intervention. The more information is exploited the more complex the process is. Exploiting this information and derive semantic relations is a core task of the ontology mapping process, and therefore it does not make much sense to use third party solutions.

Ontology instances will be typically represented in RDF. However, XML is so far the most used language in Semantic Web and it will prevail for a considerable period.

Related projects

Four distinct ontology mapping projects are considered paradigmatic approaches. (Park et al 1998) developed an extension to Protégé that consisted of a definition of the mapping between domain ontologies and problem solving methods. Different types of semantic relations are used depending on the complexity of the transformation, ranging from simple copy to functional transformations. The approach left several open points, especially concerning mapping between multiple concepts. Besides, there is no record of experiments that apply it to the Semantic Web environment. The second approach is RDFT (Omelayenko 2002), 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. The transformation of instances is limited to regular expression transformation. Thirdly, the Buster project (Stuckenschmidt and Wache 2000) 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. At last, the OntoMerge project (Dou et al 2002) adopts a combination of merging and mapping techniques. The union of the two original ontologies creates the merged ontology. Elements common to both ontologies are identified and locally unified. Bridging axioms are then specified between each of these new elements and the respective elements in original ontologies. The merged ontology can be further used as any original ontology, allowing the conversion between a third ontology and the first two ontologies. This approach is based on an inference engine, which is responsible for its poor performance. The mandatory translation of ontologies and instances to and from an internal representation might also contribute to the poor performance. The great advantage of this approach is the creation of a new ontology, allowing further mappings. However, the authors do not refer its usefulness and concrete application in real-world cases. How much ontologies can be merged while keep manageability, considering the poor performance of the system?

3. SERVICE-ORIENTED APPROACH

Ontology mapping aims to define semantic relations between source ontology entities and target ontology entities, which will be further applied at instance level to transform source instances into target instances. Semantic relations are realized through semantic bridges:

```
semanticBridge(TR, SE, TE, SC)
```

where TR is the process to be applied in transforming instances of the source entities into instances of the target entities, SE is a subset of source ontology entities considered to create the TE, the subset of target ontology entities. Finally, SC is the set of condition expressions constraining the execution of the semantic bridge.

It is virtually impossible to provide all possible transformation requirements in a centralized static ontology mapping system. This simple observation lead to the adoption of a modular, decentralized approach, where independent transformation modules are attached to the system functional core modules (e.g. bridging, execution, negotiation, evolution) (Silva and Rocha 2003a). These modules are called *Services* and comply with a specific interface, acting as an intermediary between services and functional core modules (Figure 1).

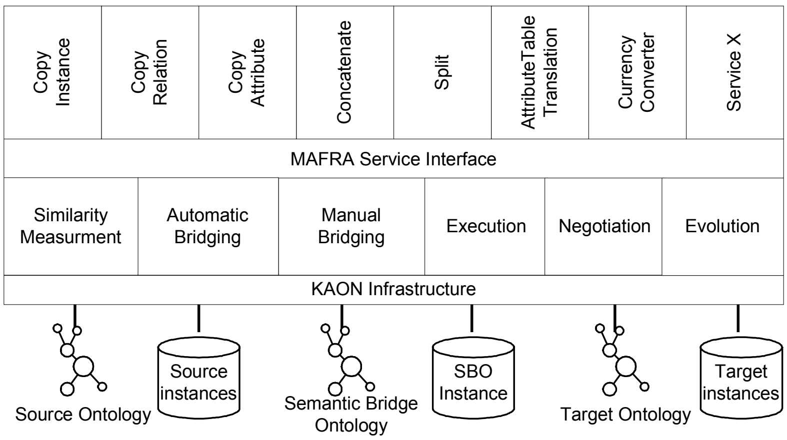

Figure 1. MAFRA Toolkit System Architecture

In the current implementation of MAFRA several transformation services are available (Silva and Rocha 2003b), providing sufficient transformation capabilities for a wide range of cases. A special distinction occurs when the *Copy Instance* service is applied in a semantic bridge. In such cases, the semantic bridge is said to be a *Concept Bridge*, and gain special characteristics (Silva and Rocha 2003b). Two characteristics are relevant (i) a Concept Bridge is a container of semantic bridges with transformation services other than Copy Instance, and (ii) Concept Bridge might form a hierarchy of Concept Bridges, following an object-oriented approach, commonly used in ontologies.

In fact, Services are responsible for many different tasks in the process, and not only for the transformations occurring at execution phase. Simple observation shows that most of the transformation process depends on the transformation capabilities, which in turn constraint all previous phases. For example, a semantic bridge is characterized by its attached service, which in turn presupposes certain arguments. It does not make sense to define arguments to a semantic bridge that are invalid (type, order, number) in the scope of the specified Service. In that sense, it is possible to validate a semantic bridge according to the attached Service.

Concluding, this approach advocates that much of the capabilities of a certain ontology mapping system are defined in Services. The goal is to exploit this premise in other phases of the ontology mapping process, relying on Services other Services-dependent tasks.

Both Similarity Measurement and Automatic Bridging are phases that might profit from this approach. While Similarity Measurement support is still under development, a prototype of the Automatic Bridging module exploiting such approach is already available in MAFRA Toolkit.

4. AUTOMATIC BRIDGING

Bridging phase is responsible for the specification of semantic bridges, where a set of source ontology entities is related to a set of target ontology entities. Such relations are based on several factors like lexical and structural similarities, leading to an overall semantic similarity. Such similarity is defined either by automatic mechanisms or by a domain expert. Consequently, the inputs of this phase are a set of semantic similarities between a pair of source and target ontologies entities, and a set of Services available for adoption.

The bridging process, whether automatic or manual, should follow the MAFRA methodology introduced in (Silva and Rocha 2003b). The proposed automatic process consists in pushing each and all similarity pairs to each available service, which in turn determines what to do with the similarity pair (Figure 2).

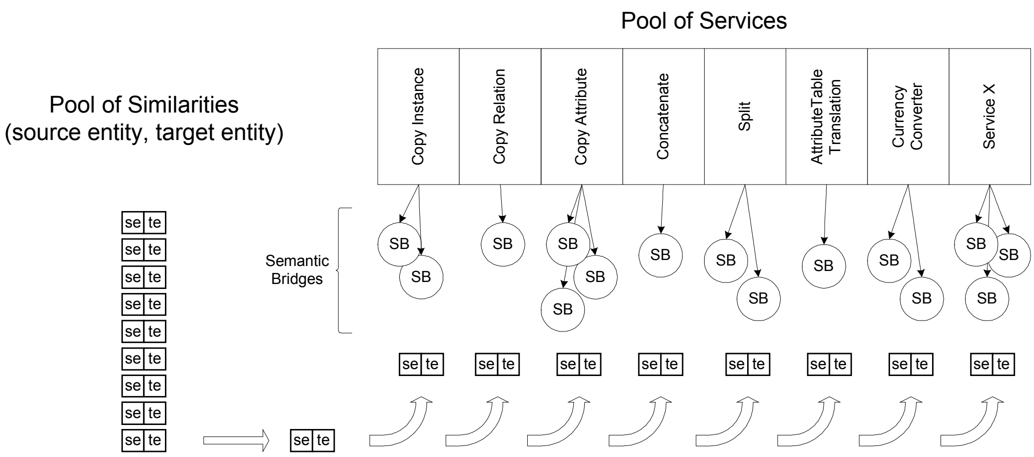

Figure 2. Figurative representation of the automatic bridging process

Three alternatives exist for each similarity pair according to service:

- It should be added to an already existent semantic bridge. In this case entities are attached to the respective arguments;
- It is relevant for the creation of a new semantic bridge. In this case a new provisory semantic bridge is created with corresponding service, and entities attached to the respective arguments;
- It invalidates an already existent semantic bridge, in which case the provisory semantic bridge is deleted.

The described process allows the same similarity pair to be applied in more then one semantic bridges. While this does not intrinsically correspond to a semantic mismatch some preventive action might be taken. Two situations were identified requiring two different approaches. First situation occurs when two semantic bridges use the exactly same set of arguments. In this case the both semantic bridges are set in scope of Alternative Bridge, which prevent the execution of both bridges and emphasizes the situation to the domain expert that can modify resulting bridges as required. Second situation occurs when a similarity pair is applied in more then one semantic bridge and therefore occurs only between property bridges. One of two distinct situations would happen in turn, whether property bridges are connected to the same or different concept bridges. Independently of the situation, user is allowed to define whether the automatic process should include alternative bridges or not. Domain expert is suggested to revise automatic mapping and is allowed to change as needed.

Two automatic bridging processes are available and used interchangeably according to domain expert requirements:

- Bridging process runs in scope of an empty semantic mapping. As consequence if a previous non-empty semantic mapping exists, the bridging process clears it, loosing all manual specification and customization of semantic bridges;
- Re-bridging process runs in scope of a previously existent non-empty semantic mapping. It preserves any manual modification or customization introduced by the domain expert, while encompassing changes in the semantic bridges arising from changes in the set of similarity pairs. Notice that manual changes in semantic mapping implicitly imply changes in the set of similarity pairs, which in turn imply changes in the semantic re-bridging again.

These two slightly different processes are necessary in order to fulfill the cyclic, iterative and interactive characteristics of an ontology mapping process advocated in scope of MAFRA.

5. EXECUTION

Execution phase is responsible for the transformation of source ontology instances into target ontology instances according to a certain set of semantic bridges, called semantic mapping, which comes up from either the Automatic or Manual Bridging phases.

The execution phase is separated in two sub-phases: (i) the creation of target concept instances, and (ii) the fulfillment of target instances properties. The first sub-phase runs according to the following algorithm:

```
foreach ConceptBridge cb in mapping {                                    (1)
   if( not cb.isAbstract() ) {                                           (2)
      instances<-getAllInstancesOf( cb.getSourceConcept() )              (3)
      table<-projectPaths( cb.getAllPaths() )                            (4)
      foreach Instance i in instances {                                  (5)
         if( passConditions( T, cb.getConditions() ) {                   (6)
            if( addInstanceToTransformationTable( id, cb, T ) ) {        (7)
               id<-cb.executeService( cb.getTargetConcept(), id ) }}}}}  (8)
```

The process runs for each concept bridge defined in the mapping (1) that is not abstract (2). All instances of source concept are identified for further transformation (3). In (4) a projection of all path values mentioned in the bridge will be performed. The projection table corresponds to the enumeration of all combinations of values accessed through the paths. Imagine the following semantic bridge where instances of Individual are transformed into instances of Man, if the value of property sex is “M” and the instance is linked to an instance of Affiliation such that its country property has “Portugal” as sub-string. Only one target instance will be created as specified by the cardinality parameter.

```
semanticBridge( CopyInstance, O1:Individual, O2:Man,
              {O1:Individual.sex=”M”,
               O1:Individual.affiliation.country Like “Portugal”,
               cardinality=1} )
```

Consider now the following instances of O1:Individual and O1:Affiliation:

```
O1:Individual(  ID=”NUNOSILVA”,
              name=“Nuno Silva”,
              sex=“M”,
              affiliation=“#ISEP”,
              affiliation=“#FZI” )
O1:Affiliation( ID=“ISEP”,                  O1:Affiliation( ID=“FZI”,
              institution=“ISEP”,                         institution=“FZI”,
              address=”Porto”,                            address=“Karlsruhe”,
              country=“Portugal”)                         country=“Germany” )
```

So, the projection table for previous instance, according to previous semantic bridge would be (Table 1):

Table 1. Projection Table example

ID	Sex	Affiliation	Affiliation.country
"#NUNOSILVA"	"M"	"#ISEP"	"Portugal"
"#NUNOSILVA"	"M"	"#FZI"	"Germany"

For each line in the table (5), the conditions are checked (6), and if verified, a new transformation occurs. In this case only the first line in the table fulfill conditions and consequently only a new instance of O2:Man is created (8). If both lines fulfill conditions, only one instance would be created, because of the cardinality constraint specified in semantic bridge. However, prior to the transformation, it is necessary to report the requested transformation to the transformation table (7). This table is responsible for the creation of a relation between source instances and target instance. It is necessary that each source instance give raise to a single target instance. If a duplicate translation is requested, an exception is raised. As referred in (Silva and Rocha 2003b), it is possible to create more then one target instance from each source instance, recurring to extensional specification of instances. Transformation table serves to keep track of identifications for further instances interrelations.

Second phase of the transformation process is responsible for the creation of target instance properties according to property bridges attached to the concept bridge (and all its super bridges) that created the instance. Next algorithm depicts the process:

```
(1) foreach Instance i in transformationTable {
(2)    cb<-i.getConceptBridge()
(3)    pbs<-cb.getAllPropertyBridges()
(4)    foreach PropertyBridge pb in pbs
(5)       pb.executeService( i ) }
```

The process runs in scope of each newly created instance (1). The concept bridge that created the instance is identified (2). The concept bridge resolves the property bridges to run over properties (3). The getAllPropertyBridges() method not only returns the immediate property bridges of the concept bridge but also all property bridges defined for super-bridges. The service associated with each property bridge is executed for the instance. The primary task for a service is to project all argument paths concerning the instance, as happened in first phase. The transformation is service dependent, but runs for each line of the projection table that passed the specified conditions and other transformation constraints, like cardinality.

6. QUERY WEB SERVICE

Even if ontology mapping might be applied in different contexts, our current efforts are focused in providing a functional system in the context of Semantic Web. We envisage an environment where autonomous agents need to transform excerpts of knowledge bases, according to momentary interactions with other agents. We advocate a transformation system centralized in a mediator responsible for the exchange of information between agents. Such mediator might be an autonomous entity or might be part of one of the interacting agents. Mediation process is preceded by a registration phase, concerned with the identification about each agent willing to participate in the community. In this phase, each agent provides self-identification (e.g. name, location), a set of ontologies it recognizes and a set of mappings it accepts, either as source or target agent. The query process runs according to the following algorithm:

```
query<-receiveQuery()
tOnto<-query.getOntology();
mappings<-getAllMappingsWithTargetOntology( tOnto )
transf<-{}
tEntities<-query.getEntities()
foreach Mapping m in mappings {
      if( areAllTargetEntitiesMapped( m, tEntities ) ) {
         cbs<-m.getConceptBridgeWithTargetEntities( tEntities )
         queryToSources<-constructQuery( m, query )
         agents<-getRegisteredAgents( m )
```

```
        sendQueryToSources( agents, queryToSources )                      (11)
        repliesFromSources<-receiveFromAgents( agents, queryToSources )   (12)
        transf<-transf+transformInstances( repliesFromSources, cbs ) }}   (13)
  reply<-runQuery( query, transf )                                        (14)
  sendReplyToAgent( query.getAgent(), reply )                             (15)
```

The mediator receives a query from an agent (1). Accordingly to the query, the mediator identifies the ontology subjacent to the query (2 and 3) and identifies all semantic mappings related to that ontology (3). Each semantic map is then traversed in order to verify if all entities referred in the query are also mentioned in the mapping (5 and 7). If so, all concept bridges that relate each of the target entities are identified (8). A new query is constructed, which will request all instances of all source concepts mentioned in all previous identified concept bridges (9). This new query is dispatched to all agents employing the source ontology in such mappings (10 and 11). The set of instances received from source agents (12) are then transformed through the previously identified concept bridges (13). When this process runs for all mappings, a set of transformed target instances exists. However, these instances might not correspond to the original query. Consequently, it is necessary to query the resulting set according to the original query (13). The result is finally sent to the requesting agent (14).

7. EXPERIENCES

MAFRA Toolkit was adopted as the development, representation and transformation engine in the Harmonise project (Harmonise 2003). Harmonise intends to overcome the interoperability problems occurring between major tourism operators in Europe. Problems arise due to the use of distinct information representation languages like XML and RDF, and different business and information specifications, like SIGRT (SIGRT 2003), TourinFrance (TourinFrance 2003). Harmonise uses the "Interoperability Minimum Harmonisation Ontology-IMHO" (Harmonise 2003) as lingua franca between agents. MAFRA is responsible for the acquisition, representation and execution of the ontology mapping between each agent specific ontology and IMHO. IMHO describes the tourism domain in about 64 concepts plus 120 attributes and 213 inter-relations between concepts. The partners ontologies vary enormously in respect to both IMHO and other ontologies. For example, the MEK ontology specifies 1 concept with 48 attributes and SIGRT defines about 50 concepts. Many different semantic and syntactic mismatches occur but no conceptual limitations were detected in the MAFRA Toolkit, and only a few refinements of the prototypal mapping relations were required.

Concerning performance issues, a very simple experience was made. Due to the lack of reported experiences comparing ontology mapping tools, the description contained in (Dou et al 2003) constitute a simple but valuable reference. Authors report the experience in transforming a dataset of 21164 instances respecting the (Gedcom 2003) ontology, into instances respecting the (Gentology 2003) ontology. These are two very similar ontologies, whose mapping requires only simple semantic relations. The MAFRA Toolkit mapping was developed according to the semantic relations presented in the report and others gathered from the transformed data set, accessed from the web. No distinctions were detectable from both transformations. Ontologies are represented in DAML, which is not directly supported by MAFRA Toolkit. However, a representation translator from DAML to RDFS is available, which transformed ontologies in a few seconds. Dataset is represented in RDF, thus excusing any transformation in MAFRA Toolkit execution. On the contrary, OntoMerge requires transformations if both ontologies and dataset. This might explain the huge difference in performance: while OntoMerge reports a 22 minutes execution time in a Pentium III at 800MHz, MAFRA Toolkit achieved the same results in less then 2 minutes in a Pentium II at 350 MHz. If a Pentium 4M 2.0Mhz is used, MAFRA requires only 1 minute and 17 seconds.

8. CONCLUSIONS

This paper puts forward a new approach to ontology mapping, based on the notion of multi-dimensional service. Such services are responsible not only for the traditional instance transformation but also for other services dependent tasks, like automatic bridging, negotiation and evolution. For the moment MAFRA

Toolkit provides support in the four modules of the MAFRA framework: lift and normalization of source ontologies and datasets, automatic and manual specification and their representation of semantic relations, instance transformation and an easy and intuitive graphical user interface.

Currently, our efforts are focused in the evolution of the ontologies and its consequences to the ontology mapping process. It is not difficult for ontology mapping to become incoherent when a number of changes occur in mapped ontologies. The adopted service-oriented architecture provides a good starting point. A longer-term project should facilitate the mapping acquisition between different agents using meaning negotiation. This phase will also potentially benefice from the service-oriented architecture, since available services would be responsible for the independent argumentation upon proposed semantic relations.

While experiences and comparisons with other ontology mapping tools are insufficient, they showed that MAFRA Toolkit fulfils real-world requirements with a good performance.

ACKNOWLEDGEMENTS

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

REFERENCES

Conference Proceeding

Dou, D. et al, 2002. Ontology translation by ontology merging and automated reasoning. *Proceedings of EKAW Workshop on Ontologies for Multi-Agent Systems.* Sigüenza, Spain.

Dou, D. et al, 2003. Ontology translation on the semantic web. *Proceedings of submitted to Semantic Web Conference.*

Horrocks, I., 1998. The FaCT System. *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux'98).* Oisterwijk, Netherlands. pp 307-312.

Maedche, A. et al, 2002. MAFRA - A MApping FRAmework for Distributed Ontologies in the Semantic Web. *Proceedings of Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at ECAI'2002.* Lyon, France. pp 60-68.

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

Park, J. Y. et al, 1998. Mappings for Reuse in Knowledge-based Systems. *Proceedings of 11th Workshop on Knowledge Acquisition, Modelling and Management (KAW 98).* Banff, Canada.

Silva, N. & Rocha, J., 2003a. MAFRA – An Ontology MApping FRAmework for the Semantic Web. *Proceedings of 6th International Conference on Business Information Systems.* University of Colorado at Colorado Springs (UCCS), Colorado Springs (CO), USA.

Silva, N. & Rocha, J., 2003b. Semantic Web Complex Ontology Mapping. *Proceedings of Web Intelligence 2003.* Halifax, Canada.

Stuckenschmidt, H. & Wache, H., 2000. Context Modeling and Transformation for Semantic Interoperability. *Proceedings of Workshop "Knowledge Representation meets Databases" (KRDB 2000) at ECAI'2000.* Berlin, Germany. pp 115-126.

Electronic Citation

Gedcom ontology, 2003. http://www.daml.org/2001/01/gedcom/gedcom.daml.

Gentology ontology, 2003. http://orlando.drc.com/daml/Ontology/Genealogy/3.1/Gentology-ont.daml.

IMHO, 2003. http://www.harmonise.org.

Ontobroker, 2003. http://www.ontoprise.de/products/ontobroker_en.

Sistema de Informação de Gestão de Recursos Turísticos (SIGRT), 2003. http://www.dgturismo.pt/irt/c_pi.asp.

TourinFrance, 2003. http://www.tourisme.gouv.fr.

OWL Web Ontology Language Guide, 2003. http://www.w3.org/TR/owl-guide/.