SERVICE-ORIENTED ONTOLOGY MAPPING SYSTEM

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 depend 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. While this paper overview MAFRA Toolkit, the main contributions and novelties are the Automatic Bridging process and the Query Web Service.

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. The ontology mapping aims to overcome semantic integration between ontology-based systems. According to the semantic relations (mapping relations) defined at conceptual level, source ontology instances are transformed into target ontology instances. 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 [1]. MAFRA covers all the phases of the ontology mapping process, naming analysis, specification, representation, execution and evolution. MAFRA Toolkit is the current MAFRA implementation. It adopts a declarative specification of mappings, hiding the procedural complexity of specification and execution, while its service-oriented open architecture allows the integration of new semantic relations into the system, improving mapping capabilities as required.

This paper is organised as follow: Section 3 presents the MAFRA Toolkit service-oriented architecture. Section 4 describes the automatic semantic bridging process while section 5 describes the query web service, which allows independent agents to interoperate based on MAFRA based ontology mapping. Section 5 presents the Graphical User Interface proposed in MAFRA Toolkit. Section 6 describes related projects and compares them with this approach. In Section 7 some experiences are described, allowing a limited perspective of the capabilities of this approach. Finally, Section 8 makes a short overview of the work done so far and points out some current and future efforts.

2 Service-oriented approach

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

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

- *TR* is the process to apply in transforming instances of the source entities into instances of the target entities;
- *SE* is a subset of source ontology entities;
- *TE* is the subset of target ontology entities;
- *SC* is the set of condition expressions constraining the execution of the semantic bridge.

It is virtually impossible to provide all possible transformation requirements using 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 (i.e. bridging, execution, negotiation, evolution, etc.) [1]. These independent transformation modules are called *Services* and provide their resources to the MAFRA core modules through the MAFRA Service Interface (Figure 1). Services are described and specified through a simple ontology, which allows MAFRA core modules to request specific features.

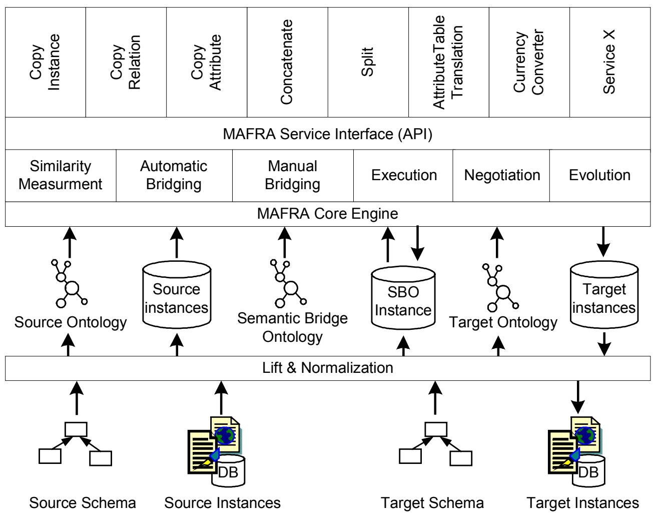

Figure 1. MAFRA Toolkit System Architecture

Simple observation shows that most of the transformation process depends upon transformation capabilities present in the system, which in turn constraint all previous phases. In fact, Services are responsible for many different tasks in the process, and not only for the transformations occurring at execution time. For example, the transformation service associated with a specific semantic bridge presumes a set of specific arguments specified in the Service description according to their characteristics (e.g. type, number, order). Recurring to this information it is possible to validate the semantic bridge arguments according to the attached Service.

The service-oriented approach suggested in this architecture advocates the need to exploit the knowledge and capabilities associated with each Service in order to increase automation and quality of the overall mapping process. Each Service interface is improved depending on the capabilities it provides to the MAFRA Core Modules. Automatic bridging and evolution are mapping phases that profit from this approach.

3 Automatic Bridging

Automatic bridging process concerns the discovery and definition of semantic bridges with minimal human intervention, based on semantic similarities between source and target ontologies entities and a set of available pool of services.

The set of semantic similarities between pairs of source and target ontologies entities play therefore a fundamental role in the discovery process. But while similarity pairs suggest that a semantic similarity exists between two entities, it says nothing about the transformation necessary to overcome the semantic heterogeneity.

It is therefore fundamental to identify the distinct sets of similarity pairs that fit in and fulfill the transformation service arguments. It is up to the service to determine the characteristics of the similarity pairs suited for its arguments. The more the service details and distinguish its arguments requirements from other services, the more perfect the automatic bridging process potentially becomes. Several and different similarity requirements are therefore required and exploited for each transformation service. Lexical and structural similarities between ontologies entities and similarities between source and target entities instances (if available) are examples of these factors. Others factors are less obvious and require deep domain expertise. In any case, similarity factors are combined into similarity pairs in the form of:

`sp(se,te,c,v)`

- *se* is the source ontology entity;
- *te* is the target ontology entity;
- *c* is the combination algorithm defined by the service;
- *v* is the value of the resulting similarity.

Each transformation service defines a threshold value for acceptance/rejection of similarity pairs. Currently, several independent similarity calculators are being developed and integrated into the system but for now the user manually defines the similarity factors.

To facilitate understanding the automatic bridging process described bellow, some basics on SBO [1] are necessary. Three types of semantic bridges exist:

- Concept Bridge: when the Copy Instance service is assigned. Concept Bridges form hierarchies of Concept Bridges, following the object-oriented approach, commonly applied in ontologies languages;
- Property Bridge: when any other service is assigned. These bridges are executed in scope of Concept Bridges;
- Alternative Bridges are containers for mutually exclusive semantic bridges.

Five steps compose the automatic bridging process:

1. The Similarity Inference step pre-processes similarity pairs to infer others. This situation occurs for all similarity pairs whose entities are relations (properties relating concepts). The problem arises due the fact that Property Bridges are defined in the scope of a Concept Bridge. This imply that domains and range concepts of properties are also semantic similar and further bridged. Consider `sp(name,surname,c1,x1)` a valid similarity pair for certain transformation service. The inference step determines that domain and range of *name* and *surname* are also semantic similar. If not previously defined, source concept *Person* and target concept *Employee* are stated as similar pair:

   `sp(Person,Employee,c2,inferred)`

   where the similarity value assumes the value *inferred*. The initial similarity pair is changed to:

   `sp(Person.name,Employee,surname,c1,x1);`
2. The Concept Bridge Specification step consists in pushing similarity pairs whose target entity is a concept, to the Copy Instance service (the service attached to Concept Bridges). For each valid similarity pair, a concept bridge is created. If the source entity is a property, the domain of property is bridged, and the property serves as extensional specification (see [2]);
3. The Property Bridge Specification step consists in pushing similarity pairs to each available service. The service itself accepts or rejects similarity pairs depending on its cardinality and similarity value (see example below). Only properties with the same domain concept are joined in a single Property Bridge;
4. The Inter-bridging step consists in creating the relations between semantic bridges. Two type of inter-bridges relations are defined: (i) Each Property Bridge is set in scope of Concept Bridges whose concepts are domains of the properties in the Property Bridge and (ii) Concept Bridges are set sub bridge of Concept Bridges whose concepts are the minimum super concept found;
5. The Alternative Bridge Specification step consists in setting certain group of bridges mutually exclusive. Similarity assignment strategy allows the same similarity pair to be applied in more then one semantic bridge. While not corresponding to a semantic mismatch, it is probable that the combination algorithm is not sufficiently distinctive from others. A special situation occurs when the exact same set of similarity pairs is used in more then on semantic bridge. In this case the process sets those bridges mutually exclusive. For that, an Alternative Bridge is created in the scope of the bridges and these are set as alternatives. The alternative bridge emphasizes the situation to the user, which is suggested to revise and customize the mapping resulting from the automatic process.

Example

Consider the automatic bridging process is trying to find the set of similarity pairs suited to the Copy Instance and Concatenation and Copy Attribute services. In its simplest form Copy Instance service takes one source concept and one target concept. The Concatenation service takes *n* source ontology attributes (strings) and concatenates their values into a single target ontology attribute (string), and Copy Attribute service takes one source attribute and copies its value to a single target attribute. After the similarity inference step, five valid similarity pairs exist:

```
sp(Person,Employee,c1,x1)                  (1)
sp(Person.givename,Employee.name,c2,x2)    (2)
sp(Person.surname,Employee.name,c2,x3)     (3)
sp(Person.givename,Employee.name,c3,x4)    (4)
sp(Person.surname,Employee.name,c3,x5)     (5)
```

where *c1*, *c2* and *c3* are the combination algorithms for the Copy Instance, Concatenation and Copy Attribute services respectively.

According to step 2, similarity pair (1) justifies the creation of a Concept Bridge (CB-Person-Employee). No other similarity pair is applicable in step 2, since no other concerns combination algorithm *c1*.

In step 3 each similarity pairs are forwarded to respective services. According to the Concatenation service cardinality, at least two different similarity pairs must have the same target attribute. Yet, attributes must have the same domain concept. Similarity pairs (2) and (3) respect these constraints and give raise to a new Property Bridge (PB-Concatenation-name). The Copy Attribute service requires that for each attribute, no other similarity pair exists (1:1 relation). Similarity pairs (4) and (5) do not respect the constraint and are therefore rejected.

In step 4, PB-Concatenation-name bridge is set in scope of CB-Person-Employee bridge, since Person is the domain concept of all source attributes, and Employee is the domain concept of the target attribute.

Step 5 has not effects once the previously created bridges correspond to completely different set of similarity pairs.

3.1 Bridging vs. Re-bridging

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.

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

Manual creation or deletion of semantic bridges implicitly implies changes in the set of similarity pairs. Such semantic bridges changes are not incorporated into the similarity pairs view unless requested.

4 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()                              (1)
tOnto<-query.getOntology();                        (2)
mappings<-getAllMappingsForOntology(tOnto)         (3)
transf<-{}                                         (4)
tEntities<-query.getEntities()                     (5)
foreach Mapping m in mappings {                    (6)
  if(areAllEntitiesMapped(m,tEntities)){           (7)
    cbs<-m.getCBsWithEntities(tEntities)           (8)
    query<-constructQuery(m,query )                (9)
    agents<-getRegisteredAgents(m)                (10)
    sendTo(agents,query)                          (11)
    replies<-receiveFrom(agents,query)            (12)
    transf+=transformInst(replies,cbs)}}          (13)
reply<-runQuery(query,transf)                     (14)
sendTo(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 one 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 one of the mappings (10 and 11). The set of instances received from source agents (12) are then transformed through the previously identified concept bridges (13). After the cyclic process of lines (6) to (13), a set of transformed target instances is available. It is now necessary to run the original query against the set of resulting instances to filter instances accordingly to query (14). The result is finally sent to the requesting agent (15).

5 Graphical User Interface

The graphical user interface provides the domain expert with an extensive set of functionalities to specify and customize the semantic mapping and similarity pairs. The GUI evolved to a fully graphical representation of ontologies and semantic mapping (Figure 2), exploiting the graphical support available from KAON [3]. Using the same graphic approach the user is allowed to customize the mapping, matches or ontology entities. Each entity is represented using different shapes and colors.

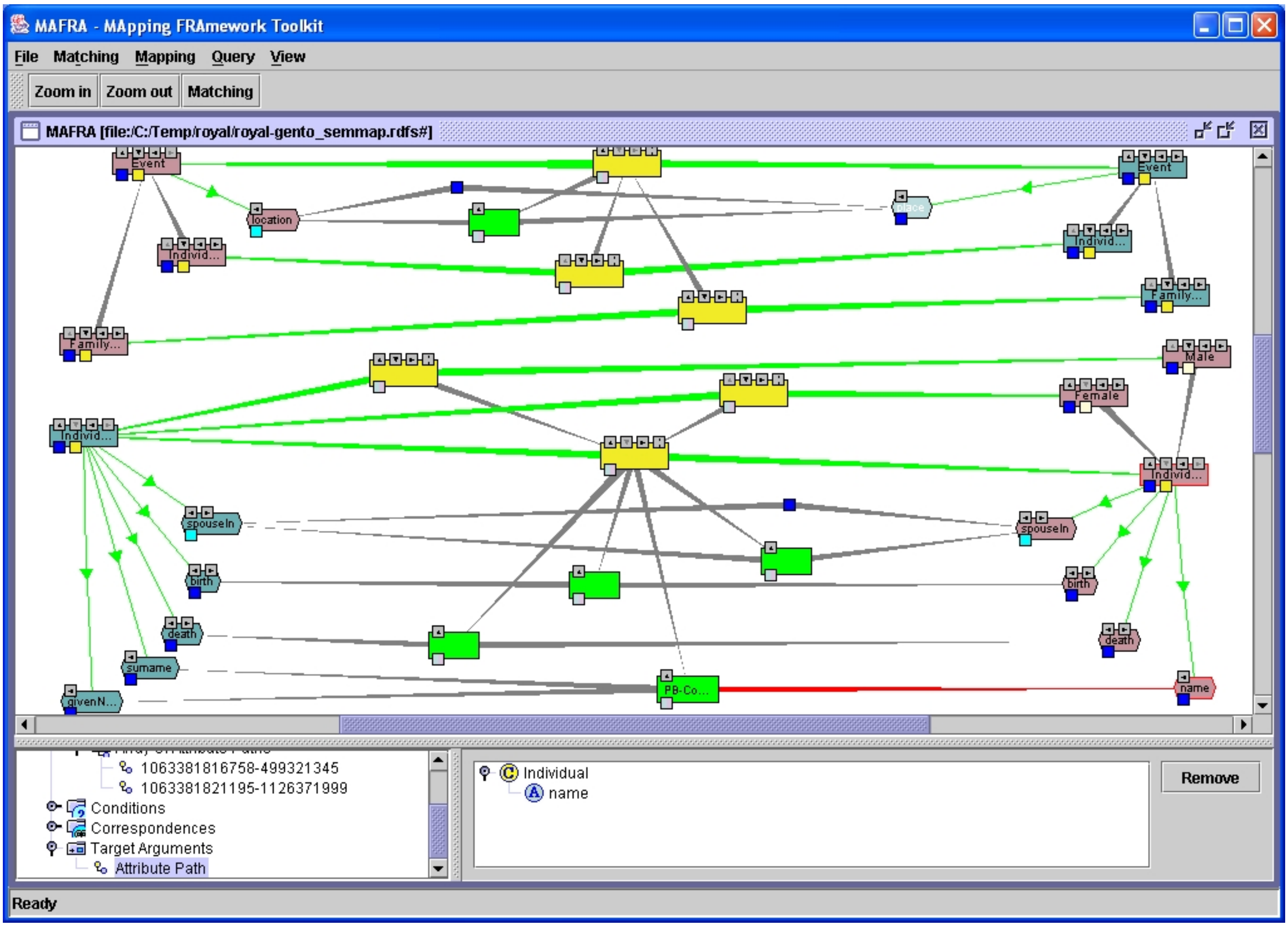

Figure 2 - MAFRA Toolkit screen-shot: specification or/and customization of semantic bridges

Ontology entities are connected to semantic bridges and similarity pairs using mouse-operated connections. Entities specific information is presented in the lower part of the window, allowing definition or customization of arguments, otherwise inaccessible. In special, it is necessary to specify transformation elements that do not exist in ontologies, like string separators, string patterns, currency converter factors, etc.

It is possible to manipulate the inter-relations of both ontologies and mapping elements, using the small colored control buttons in the border of the elements. Each button has two states (on/off) that expands or collapse the respective connections. Context menu with function like Hide, Hide Others and Fix nodes, permits to extensively define what is presented in the GUI.

6 Related Projects

Four ontology mapping projects are considered paradigmatic approaches. Park et al. [4] describes 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 [5], 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 [6] 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. The OntoMerge project [7] 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 defined. 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?

7 Experiences

MAFRA Toolkit was adopted as the development, representation and transformation engine in the Harmonise project [8]. This project 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 (http://www.dgturismo.pt/irt/c_pi.asp), TourinFrance (http://www.tourisme.gouv.fr) and FTB (http://www.mek.fi). Harmonise uses an "Interoperability Minimum Harmonisation Ontology" 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, 120 attributes and 213 inter-relations between concepts. IMHO and partner ontologies are very different. 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 MAFRA Toolkit, and only a few refinements of the prototypal mapping relations were required.

Concerning performance issues, a very simple experience was made. Considering the lack of experience reports with ontology mapping tools, the report contained in [9] constitute a simple but valuable reference. They report the experience in transforming a dataset of 21164 instances respecting the Gedcom ontology (http://www.daml.org/2001/01/gedcom/gedcom.daml), into instances respecting the Gentology ontology (http://orlando.drc.com/daml/Ontology/Genealogy/3.1/Gentology-ont.daml). 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 transform 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 1 minute and 17 seconds. Unfortunately, it was not possible so far, to formally evaluate performance of the system.

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, relying on services the 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.

9 Acknowledgements

This work is partially supported by the Portuguese MCT-FCT project POCTI/2001/GES/41830 under FEDER. Thanks to Jorge Santos for their comments and revisions.

10 REFERENCES

1. Maedche A., Motik, B., Silva, N., and Volz, R.; "MAFRA - A MApping FRAmework for Distributed Ontologies in the Semantic Web"; Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at ECAI'2002; Lyon, France; 2002.
2. Silva N. and Rocha, J.; "Semantic Web Complex Ontology Mapping"; Web Intelligence 2003; Halifax, Canada; 2003.
3. Motik B., Maedche, A., and Volz, R.; "A Conceptual Modeling Approach for building semantics-driven enterprise applications"; ODBASE-2002; California, USA; 2002.
4. Park J. Y., Gennari, J. H., and Musen, M. A.; "Mappings for Reuse in Knowledge-based Systems"; 11th Workshop on Knowledge Acquisition, Modelling and Management (KAW 98); Banff, Canada; 1998.
5. Omelayenko B.; "RDFT: A Mapping Meta-Ontology for Business Integration"; Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at ECAI'2002; Lyon, France; 2002.
6. Stuckenschmidt H. and Wache, H.; "Context Modeling and Transformation for Semantic Interoperability"; Workshop "Knowledge Representation meets Databases" (KRDB 2000) at ECAI'2000; Berlin, Germany; 2000.
7. Dou D., McDermott, D., and Qi, P.; "Ontology translation by ontology merging and automated reasoning"; EKAW Workshop on Ontologies for Multi-Agent Systems; Sigüenza, Spain; 2002.
8. Harmonise; IMHO; http://www.harmonise.org; 2003.
9. Dou D., McDermott, D., and Qi, P.; "Ontology translation on the semantic web"; ODBASE-2003; Catania (Sicily), Italy; 2003.