XeOML: An XML-based extensible Ontology Mapping Language[*]

Maria Teresa Pazienza[1], Armando Stellato[1], Michele Vindigni[1],
Fabio Massimo Zanzotto[2]

[1] DISP- University of Rome Tor Vergata
Via del Politecnico, 1 – 00133 Roma ITALY
{pazienza, stellato, vindigni, zanzotto}@info.uniroma2.it
[2] DISCO - University of Milano Bicocca
P.zza Ateneo Nuovo 1 – 20126 Milano ITALY
zanzotto@disco.unimb.it

Abstract. Semantic Interoperability is a crucial issue in the Semantic Web context: web services and portals, providing real-time access to widely distributed information sources, need to overcome problems due to heterogeneities in the use of distinct locales, languages and idioms. Though standardization efforts for semantic content representation are converging toward some concrete standards, still a lot of work is required to achieve real interoperability over knowledge content as managed by communities of autonomous and distributed individuals. An unavoidable trade-off between coverage of heterogeneous information sources and achievement of common semantics for accessing their content emerges. In this framework, we have carried out our research to develop an extensible language (XeOML) for describing mappings between domain ontologies, in which knowledge representation formalisms and similarity measures can be dynamically added according to community needs and intent.

1 Introduction

The goal of granting semantic accessibility to the web content pursued by the Semantic Web [6] can be achieved through ontologies, as they play a crucial role in supporting the exchange of data, in providing a formal vocabulary for the information and in unifying different views of a domain in a safe cognitive approach [9].

Despite all the active researches on ontology reuse, the inherently decentralized nature of the WWW pushes for a multitude of autonomously conceived choices, where different communities adopt their locally developed ontologies to represent their own knowledge. Different solutions to the knowledge sharing issue have been proposed and experimented: each of them is related to a different framework, thus suggesting a coarse classification with respect to the constraints they impose on the way knowledge must be represented and structured. Many existing information systems exploited both as research results (TSIMMIS [1], Information Manifold [2], Infomaster

---

[*] This work has been partially funded by the EU project MOSES IST-2001-37244.

[3], MOMIS [7]) or industrial solutions (Xyleme[1]) proposed centralized systems of mediation between users and distributed data sources, which exploit mappings between a single mediated schema and the local schemas. Other approaches (such as Mafra [8]) followed a more flexible solution based on distributed mediation systems. These systems generally include static representations of the relationships that bind different and distributed knowledge resources, or in some cases rely on the behavior of underlying communities of software agents [9,10,12] to dynamically negotiate the meaning of both concepts and relations from the different ontologies.

We will not analyze here the pros and cons of such approaches, being out of the scope of this work; by looking at the scenarios depicted above, we will highlight that, whichever the situation to be considered, either sharing knowledge between myriads of (couples of) independent ontologies or linking every local ontology to a centralized one, a key issue for the completion of real knowledge interoperability is represented by the definition of appropriate mappings between ontologies. We will focus on such a vital problem for any application scenario.

We refer to the ontology mediation activity as the process of reconciling differences among different information sources (and their schemas), to achieve interoperability between several applications and their underlying annotated data. This activity includes “discovery” of ontology mappings, that is, of declarative specifications of the semantic overlap between two (or more) ontologies. The mappings can broadly vary depending on the tasks they will support: different scenarios could require either *injective (*specifying how to go from a source to a target ontology) or *bijective* (stating equivalences among concepts and relationships in both ontologies) correspondences, different accuracy in establishing semantic similarities, and different levels of coverage of the mapped information sources. These differences are often underestimated in literature where most researches are devoted to define languages that completely state correspondences between entities, mixing together declarative and operational aspects of this task.

We argue instead that several factors interact in real world scenarios: complex mappings and reasoning capabilities are both necessary for comparing and combining ontologies and for integrating the data they describe. A big effort has been devoted in defining knowledge representation languages powerful enough to express the different views of a domain. OWL [14] has recently been accepted as a W3C recommendation for the representation of ontologies on the Web, and this represents an important step towards the realization of the Semantic Web vision. Far beyond the standardization of knowledge representation however, still remains the problem of reaching semantic consensus at content level. In the context of an open environment, what really happens is that many different heterogeneous ontologies with overlapping domains exist and may be shared by several partners of the communication.

To cope with such heterogeneity there is a need for tools and languages to formally and explicitly specify ontology mappings in order to achieve the desired interoperability. As pointed out in [15], OWL offers limited support for these mappings through the import statement that is used to import an ontology into another one: after importing, relationships among concepts in the different ontologies can be specified through equivalence and subsumption axioms.

---

[1] http://www.xyleme.com/

However, the mechanisms provided by OWL can reveal unsatisfactory for mapping specification even in the general case. OWL promotes a tight coupling between ontologies, as it makes dependent the importing ontology on the imported one. In a dynamic scenario, where ontologies should be updated to reflect changes in the domain, this kind of dependence does not allow for a flexible knowledge organization, and can result in a severe loss of consistency that usually is very difficult to be amended. Moreover, in [15] is also argued an epistemological inadequacy of OWL as a mapping language, because Description Logic constructs in OWL are useful for describing merged ontologies while general ontological mappings are not supported.

In [11] a significant extension to the OWL model has been proposed in the form of C-OWL, a language which allows for the representation of “contextual ontologies”, intended as OWL ontologies embedded in a space of other OWL ontologies and related to them via context mappings [5]. Inside that work, five bridging rules, which account for four levels of similarity (identity, generalization/specification, compatibility and orthogonality) are defined to map concepts between different ontologies. The above rules do not take into account, however, of the really complex relationships which may hold between ontological entities of different types or even involving complex structures of entities from any of the mapped resources. A trade-off between generality and adaptivity of the proposed mapping model on the one side, and accurateness and completeness towards every possible contingency is however hard to balance. To this end, a deeper introspection inside the knowledge representation models which are mostly adopted nowadays and the factorization and formalization of the recurring constructs is throughout necessary, in order to obtain a language which can be tuned to different situations, still maintaining the integrity of its underlying fabric.

In the remainder of this paper, we present a novel mapping specification language, XeOML, that bases on a layered approach to define a well formed declarative representation of complex structural correspondences between the entities involved in a mapping process. Formal semantics of the core layer of the language is intentionally omitted, as we aim to keep it separated from the more structural aspects of the problem. The syntax of the language has in fact been chosen to make possible an incremental specification of most accurate correspondences, in order to leave such details to the specific situations arising in application dependant contexts (where the actual reasoning capabilities and the expressiveness of the formalisms used to represent the knowledge become clear, making easier to define more effective ways to deal with such aspects).

2 XeOML: An XML-based extensible Ontology Mapping Language

XeOML is an extensible language for describing ontology mappings, developed at the University of Roma Tor Vergata and adopted by first for the ontology mapping task inside the European project Moses[2]. As its acronym suggests, it is based on XML

---

[2] Examples in the rest of the paper refer to MOSES project that deals with a distributed multilingual Q/A system (based on ontological knowledge) accessing to different university web sites.

syntax, taking advantage from its expressive power to offer a core language characterized by easy machine-readability and high extensibility.

XeOML is defined by an XML schema[3], *AbstractMapping*, which provides information for describing mappings between ontologies, detailing the structure of a mapping document and defining the set of elements that populate an ontology.

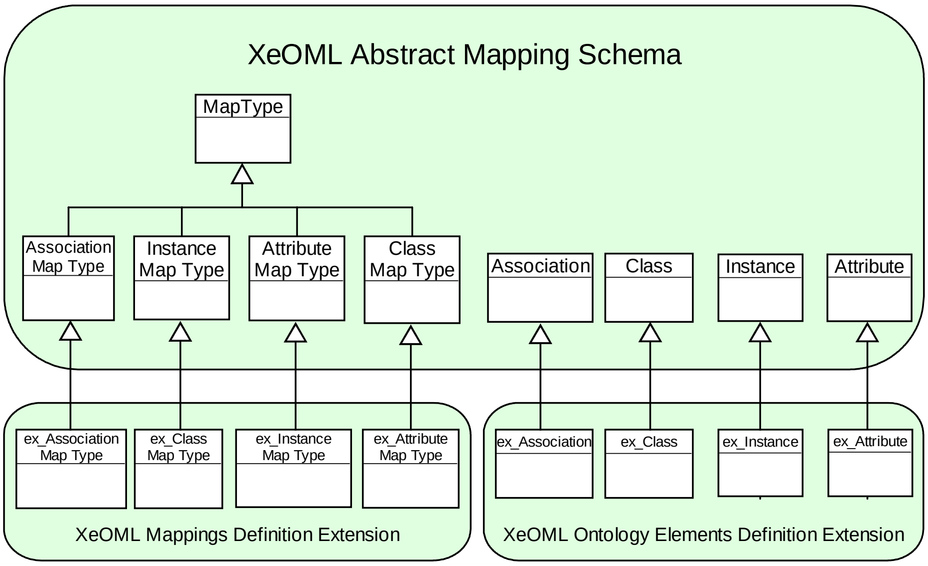

Fig. 1 An overview of the Abstract Mapping Schema and its Extensions

The *AbstractMapping* schema (Fig. 1) thus defines a core language for ontology mapping representation, voluntarily ignoring details on different levels of mapping relationships which may be considered among ontology elements and on the semantics associated to them: what is clearly asserted and organized in this schema, is the declaration and classification of typical *mapping patterns* that may involve complex structures of entities from the ontologies to be mapped. More semantically declarative information may thus be plugged to the main schema in the form of XML Schema extensions, which reflect different perspectives and approaches to the mapping process and/or heterogeneous knowledge representation styles; the core schema, together with its extensions, forms a complete mapping document definition.

The two extensions which need to be provided are:

- an *Ontology Elements Definition* schema extension, which accounts for specification of ontological elements according to a given representation language
- a *Mappings Definition* schema extension, where ad-hoc descriptions of the level of mappings that may be considered and agreed inside a particular framework may be specified.

---

[3] XeOML Abstract Mapping Schema and its extensions are freely available for download at: http://ai-nlp.info.uniroma2.it/xeoml

Thanks to this approach, the agents that want to exchange knowledge inside a distributed framework may rely on the same basic functionalities for interpreting the core language (thus favoring reuse of existent technologies) and need only to be “tuned” to the extensions adopted inside their community, in order to capture and exploit the committed semantics for both recognizing and ranking ontology mappings.

In Fig. **1** a partial overview of the *XeOML Abstract Mapping* Schema is given, reporting only the XML element types which are defined as abstract, thus needing to be implemented in the two schema extensions.

In the next paragraphs, a brief description of the structure of a mapping document, as implied by the *AbstractMapping* schema, will be given.

Mapping Terminology

As an ontology mapping language, XeOML foresees the presence of two target ontologies that need to be mapped. In the rest of the paper we will address these two ontologies as *Left Ontology* and *Right Ontology* (*LO* and *RO*, respectively). The *AbstractMapping* schema defines an ontology as composed of four different ontological entities: *instances*, *classes*, *properties* and *associations*; these elements reflect most of the more common ontological definitions, like those proposed by OWL [14], OKBC [4] or Topic Maps [13]. The syntax definitions for these four elements will be described in the *Ontology Elements Definition* schema, according to the model adopted to represent the knowledge content.

Two types of mappings are defined inside the *AbstractMapping* schema:

- *Simple Mappings* (or, simply, *mappings*), i.e. one-to-one relations between ontology elements of the same type.
- *Complex Mappings*, i.e. mappings involving even more than one element from one or both the ontologies; different ontology element types may be correlated into heterogeneous combinations, depending on the specific mapping relation.

We will use the term *Mapper*, indistinctly referring either to an automatic process for both producing ontology mappings or managing a meaning negotiation activity, or to a human annotator who will produce a manual mapping between the two ontologies

Mapping Structure

The structure of the mapping task is very complex in its nature. To obtain a uniform management of mappings between elements from the two ontologies, every ontology element from both *LO* and *RO* must always be included in a (simple) *mapping*, as this kind of mapping can be characterized by one the following:

- a *one-to-one* correspondence between two ontology elements
- a single element from one of the two ontologies and a reference to a complex map, meaning that the given element is involved in that complex mapping

An automatic process willing to know how an ontology element is mapped, only needs to inspect (in a uniform way) simple mappings, and, where necessary, be redirected towards a complex map; see **Ex. 1** where the participation of the “Professor” Class from *LO* in a complex map is reported.

```
<mapping xsi:type="absm:ClassMap" ID="c2">
    <MapRank xsi:type="map:ClassMapType">ExtensionalEquivalence</MapRank>
    <LeftMapped>
        <Class xsi:type="oed:OWLClass" ID="Professor"/>
    </LeftMapped>
    <RightMapped>
        <participationInMapping xlink:href="#cc1"/>
    </RightMapped>
</mapping>
```

Ex. 1 a class participating in a complex map

Notice the element inside the LeftMapped tag: it is a class as defined in OWL, because the *Ontology Elements Definition* schema extensions which implements OWL definitions has been adopted to represent this class in the example.

The Abstract Mapping Schema defines an abstract ComplexMapType (so that even this aspect can be extended to meet specific requirements) and its subclasses with some concrete types for diverse kind of Complex Mappings. We analyze here some of these types:

AttributeAggregationMap: it represents a map between one attribute from one of the two ontologies and more than one attribute from the other ontology.

```
<complex_mapping xsi:type="absm:AttributeAggregationMap" ID="cc2">
    <MapRank xsi:type="map:AttributeMapType">RangeEquivalence</MapRank>
    <MappingFuctional xsi:type="map:AttrAggMapType">StringConcat</MapRank>
    <LeftMapped xsi:type="oed:OWLDatatypeProperty" ID="name"/>
    <RightMappedAggregation>
        <Attribute xsi:type="oed:OWLDatatypeProperty" ID="name">
            <label xml:lang="en">first name</label>
        </Attribute>
        <Attribute xsi:type="oed:OWLDatatypeProperty" ID="surname">
            <label xml:lang="en">last name</label>
        </Attribute>
    </RightMappedAggregation>
</complex_mapping>
```

Ex. 2 aggregation of attribute values

An example of this map is given in **Ex. 2**, reporting string concatenation of more attributes into one, e.g. attribute *name* from a *LO* is mapped to the concatenation of *name* and *surname* from *RO*. Other cases could include attributes from one ontology whose ranges correspond to the union of the ranges of different attributes from the other. All these cases should be shown in the Mapping Definition extension and explicit semantics for handling them should be captured by agents responsible for ontology mediation activity.

ClassAggregationMap: in **Ex. 3** two classes coming from *LO*, “man” and “woman” are mapped to the class “human” from *RO*. This assertion implies that the classes “man” and “woman” can be considered as complete partitions of the class “human”. ClassAggregation Complex Mappings should deal with this sort of relationship.

```
< complex_mapping xsi:type="absm:ClassAggregationMap" ID="cc1">
    <MapRank xsi:type="map:ClassMapType">ExtensionalEquivalence</MapRank>
    <LeftMappedAggregation>
        <Class xsi:type="oed:OWLClass" ID="woman"/>
        <Class xsi:type="oed:OWLClass" ID="man"/>
    </LeftMappedAggregation>
    <RightMapped xsi:type="oed:OWLClass" ID="human"/>
</complex_mapping>
```

Ex. 3 aggregation of Classes

Istance-ClassMap: very often, depending on the conceptualization of the world and on the objectives that lies behind the development of an ontology, the same concepts appears either in the form of a class or of an instance. Theorically, a class is a “set of instances” and could never be compared to an instance, as clearly motivated in [14]. On the other hand, this is not in line with several typical ontology modeling approaches where a concept is conceived as an instance or a class depending either on the given level of abstraction or on the task the ontology is thought for.

```
<complex_mapping xsi:type="absm:ClassWRestr-ClassMap" ID="crc1">
    <MapRank xsi:type="map:ClassMapType">ExtensionalEquivalence</MapRank>
    <LeftMappedClass xsi:type="oed:OWLClass" ID="automobile_rossa">
        <label xml:lang="en">red car</label>
    </LeftMappedClass>
    <RightMappedClassWithRestrictions>
        <Class xsi:type="oed:OWLClass" ID="car"/>
        <AttributeRestriction>
            <Attribute xsi:type="oed:OWLDatatypeProperty" ID="color"/>
            <Restriction>red</Restriction>
        </AttributeRestriction>
    </RightMappedClassWithRestrictions>
</complex_mapping>
```

Ex. 4 A class mapped to another class with a value restriction on one of its attributes

ClassWithRestrictions-Class and ClassWithRestrictions-Instance: these two kind of mappings deal with conceptual equivalences between classes (instances) and partitions of classes which depend on restrictions over the range of one or more of their attributes. In **Ex. 4**, the class “red_car” is extensionally equivalent (in the sense of: share the same instances) to the class car with a restriction on the range of its “color” attribute set to “red”.

3 Extending the XeOML Schema: a case study

We hereafter describe, as an example of possible extensions, two schemas for the XeOML language, providing respectively:

- definitions for OWL ontology elements
- enumerated descriptions of possible distinct levels of conceptual similarity, classified depending on mapping type (i.e., the type of elements involved in a mapping)

OntologyElementsDefinition Schema: implementation for OWL

As a first possible extension to the XeOML language, we provided implementations for all of the XeOML abstract Ontology Elements in the form of OWL data types. All defined elements contain an ID attribute for specifying the ID of the concept from the ontologies, and an optional number of labels to represent these concepts in different languages, as defined for almost all OWL categories. We stress here that it has not always been a straight 1-1 mapping between XeOML abstract types and elements from the implemented model: in the OWL case, both OWL DataType properties and OWL Object properties have been mapped as XeOML Attribute types.

On the contrary, being Associations not explicated in OWL, they are represented in XeOML by normal OWL Classes (many knowledge representation languages do not allow for associations, being them mimed by classes, with attributes acting as roles of the association: this way of modeling is typically indicated as Association Classing).

The idea behind the extensible definitions of element types to different representation formalisms, is that a mediation activity involving two agents, requires them to be only proficient about the knowledge model adopted to express their underlying ontological resource while not necessarily being able to understand the model owned by the interlocutor. This way every agent could fully exploit the detailed semantics of its knowledge model, and leave as meaningless strings the concepts expressed for the other ontology in the mapping document, as they need only to be used as a transaction mechanism inside the mediation activity. The choice of allowing for detailed and language dependent descriptions of the ontological elements instead of neutral IDs (which could be of help in retrieving the same information from the source ontologies), may be questionable. However, although it is introducing redundancy inside the mapping document, it is indeed true that an agent exploiting this so-defined IDs would need the capability of matching them with elements from the source ontology (this may not be always trivial). Moreover, there could be many reasons to introduce more information in the ontology elements definition schema, which could be useful for making fast inference over large data from the mapping document as a whole, without the need for explicit reference to the source ontologies.

MappingsDefinition Schema

The *Abstract Mapping Schema* declares four types of mappings, related to the four basic ontology elements types: *InstanceMapType*, *ClassMapType*, *AttributeMapType* and *AssociationMapType*. A *MappingsDefinition* Schema Implementation should offer enumerated restrictions to these types, assigning specialized semantics to the level of similarity between elements of the mapped ontologies. We have produced a *Default MappingDefinition* Schema extension, providing a few examples of possible level of mappings which could be reported in a mapping document. The intent of these mapping types is to specify at what extent the knowledge data (classes, attributes and instances) that is available in an ontology can be augmented with the foreign data contained in other ones.

DefaultInstanceMapType is a restriction of the generic *InstanceMapType* and foresees the following levels of similarity between ontology instances:

1. *Equivalence*: two instances are equivalent if they refer to the same object of the world. For example "President of USA Bush" and "George W. Bush" are, apart from their different surface forms, probably referring to the same person, intended as a unique individual (in the hypothesis we are not speaking of another person with the same name as the U.S. President!).
2. *Similarity*: the two instances represent very similar concepts, though cannot be considered, under all aspects, as totally overlapping.

DefaultClassMapType is a restriction of the generic *ClassMapType* and foresees four levels of similarity between ontology instances:

1. *ExtensionalEquivalence*: it is hard to tell if two concepts are totally equivalent; many a knowledge theory should even confute the notion of equivalence between concepts coming from two different agents (either automatic agents accessing ontologies to convey the meaning of their knowledge, or humans involving in discourse). Nevertheless, there are objects and individuals in the world which may be considered unique and to which unambiguously refer. If we consider the extension of a class as the list of objects/instances referred by it, then we may say that two classes share a *Extensional Equivalence* if they describe the same instances of the world. What can be inferred from such a mapping type is that, given two classes A and B, extensionally equivalent, every instance of A may be considered an instance of B too, and vice versa. In some cases, these instances may result to be super specified or under specified, depending on the intensional similarity of the two classes: if A has some attributes which have no equivalent in B, instances of B would be under specified wrt these characteristics; this is indeed a partial lack of knowledge that cannot be filled otherwise.
2. *IntensionalSimilarity*: this level of class-similarity holds when it is not sure two given classes share exactly the same instances, but indeed they share deep intensional similarity, expressed through different aspects (terminological affinity, structural similarity, common instances and so on…). This kind of similarity does not guarantee any strong semantic implication, though could be useful in some contexts: a query to a QA system could benefit of "similar" matches, as the retrieved information may then judged by the human receiving the answer.
3. *SuperClass-Of (SubClass-Of)*: it holds when the class from *LO* represents a more general (specific) concept than the one from *RO*. The term SuperClass-Of (SubClass-Of) indicates that the class from *LO* could be ideally considered as a SuperClass (SubClass) of the one from *RO,* should the two ontologies be merged. The semantics follows as for ExtensionalEquivalence: If a class A is SuperClass

of a class B, it is possible to consider each instance of B as an instance of A, though, in this case, the converse is not true.

DefaultAttributeMapType: it restricts the abstract *AttributeMapType* with the similarity cases between ontology attributes defined below:

1. *RangeEquivalence*: should the range of two considered attributes (in their original definition, not considering restrictions applied to them when they are attached to different classes) be covering elements which are themselves mapped as equivalent, then a *RangeEquivalence* is considered for them.
2. *RangeSimilarity*: should the range of two considered attributes (in their original definition, not considering restrictions applied to them when they are attached to different classes) be covering elements which are themselves mapped as similar, then a RangeSimilarity holds between them.
3. *RangeTypeMismatch*: should the range of two considered attributes (in their original definition, not considering restrictions applied to them when they are attached to different classes) be covering elements which present a mapping mismatch of any kind, then a RangeTypeMismatch holds between them.
4. *RoleInversionRangeTypeMismatch, RoleInversionRangeSimilarity, RoleInversionRangeEquivalence*: these three types of matching are under all aspects equivalent to the (corresponding) first three ones, with the exception that Range and Domain and inverted. A match of this kind should be reported if a direct match (i.e. a match from one attribute to another attribute with proper Domain and Range equivalences) is not available for the desired attribute.

In this schema, we limited specification of attribute similarities only to range similarity, as diverse knowledge representation languages do not allow for explicit specification of attributes' domain. Association similarity levels are still under study.

Manual Mappings between Federated Ontologies: some considerations

Mapping procedure, as it has been previously described in details, may be carried on also by humans. Mappings should be defined at the best of *Mapper*'s knowledge; this implies that humans producing a map should base on their personal knowledge and perspective of the world, together with observations on structural similarities between the two given ontologies. An *Automatic Mapper* could instead exploit some external linguistic/ontological resources to obtain a wider knowledge in judging mappings between the domain ontologies.

The two main considerations to be clearly assumed when producing a map are:

1. *avoid complex mappings wherever simple mappings are available*
2. *avoid "low-rank" mapping types when "high-rank" mappings are available*

Regarding the first assertion., it is important to clarify that, while complex mappings are a valuable mean to represent complex conceptual anchors between elements from different ontologies, they should only be used to fill "holes" in the mapping document that could never be bridged otherwise.

The same considerations hold for the second statement, as it is best to map a class/instance/attribute/association with its best matching counterpart and omit the (many) other degraded mappings that may involve the given ontology element. So, for example, if a given class C_L from *LO* is extensionally equivalent (the highest rank of available class mappings) to a class C_R fro *RO*, C_L and C_R must be mapped using the proper map type without caring about other classes from *RO* that may be “more or less” similar to C_L; for the same reason, if an attribute A_L from *LO* is in *RangeEquivalence* with an attribute A_R from *RO* and if there exists an attribute I_R in *RO* which is the inverse role of A_R, there is no need of stating the *RoleInversionRangeEquivalence* between A_L and I_R: such a strategy is necessary to prevent an exponential growth in the range of possible relations that should be instantiated for every given ontological entity. If an element has no direct counterpart, it is up to the mediating agents to choose the best strategy for navigating their own ontologies and look for conceptual correspondences with the other ones: this is a fundamental difference wrt MAFRA [8], where a mapping ontology is built to bridge every (mappable) element from the two ontologies, or C-OWL [11], where the provided examples show complete mappings using the given primitives. Our future research work will go in the direction of formalizing conditions for completeness and compactness of a mapping document, to enhance the quality of automatically inferred mappings.

4 Conclusions

Ontologies are very often considered as a mean to conceptualize and express (by means of concepts and relationships) all of the knowledge relevant to a given domain, thus supporting automatic reasoning.

Nowadays, although still retaining their original role, Semantic Web has pushed forward the idea of ontologies as a mean for enabling knowledge sharing and reuse. Semantic Interoperability is thus a crucial issue in scenarios where different and distributed resources need to be reconciled, overcoming heterogeneities rising from distinct locales, languages, and, at a deeper analysis, different ways of structuring and organizing the information knowledge.

Following this trend, we are now facing a growing need for tools, languages and formalisms that should support the sharing of domain knowledge in a wide variety of different situations. XeOML, with its layered approach, tries to suggest a new direction for the representation of mappings between ontologies. This language presents a simple unifying view over the kind of elements that should be considered as relevant in every ontological framework, along with a classification of the diverse relationships that may occur between them in unforeseen mapping scenarios. Several extensions can be specified over the core language, providing local and more specific descriptions of the mapped ontological elements and detailing with the desired accuracy the relationships they are involved in.

With approaches like that, the unavoidable trade-off laying between coverage of heterogeneous information sources and preservation of common semantics to access the related content can thus be coped with: under a social perspective, communities which aim to reach knowledge sharing and services interoperability, can rely on a well assessed formalism for mapping their knowledge, only needing to commit to the agreed semantics for qualifying and ranking knowledge mappings (XeOML Mapping Rank Language Extension); at the same time, if we consider the technological aspect of the approach, the distributed agents which are meant to exchange knowledge may rely on the same basic functionalities for interpreting the core language and need only to be “tuned” to the community, in order to capture and exploit the committed semantics for recognizing and ranking ontology mappings.

Acknowledgements This work has been partially funded by the EU project MOSES IST-2001-37244.

References

1. Hammer, J., Garcia-Molina, H., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom, J.: Information Translation, Mediation, and Mosaic-Based Browsing in the TSIMMIS System. In Exhibits Program of the Proceedings of the ACM SIGMOD International Conference on Management of Data, page 483, San Jose, California (June 1995).
2. Levy, A. Y., Rajaraman, A. and Ordille, J. J.: Querying Heterogeneous Information Sources Using Source Descriptions. In Proceedings of VLDB-96 (1996)
3. Genesereth, M. R., Keller, A. M. and Duschka, O.: Infomaster: An Information Integration System. In proceedings of 1997 ACM SIGMOD Conference (May 1997).
4. Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P.D. and Rice, J. P.: OKBC: A programmatic foundation for knowledge base interoperability. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), pages 600-607, Madison, Wisconsin, USA, MIT Press (1998).
5. C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. Artificial Intelligence, 127(2):221–259, April 2001
6. Berners-Lee, T., Hendler, J., and Lassila, O.: The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American. (May, 2001).
7. Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.: The MOMIS approach to Information Integration. IEEE and AAAI International Conference on Enterprise Information Systems (ICEIS01), Setúbal, Portugal (7-10 July, 2001).
8. Maedche, A., Motik, B., Silva, N. and Volz, R.: Mafra a mapping framework for distributed ontologies. In Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW-2002, Madrid Spain (2002).
9. Pazienza, M.T., Vindigni M.: Language-based agent communication. 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02), Sigueza, Spain (2002)
10. Pazienza, M.T., Stellato, A., Vindigni, M.: ALINAs: un'architettura multilayer ad agenti per il supporto alla comunicazione linguistica. Dagli oggetti agli agenti (WOA2002), Milano, Italy (2002)
11. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L. and Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies, ISWC'2003, LNCS 2870, Sept. 2003, pp. 164--179
12. Pazienza, M.T., Vindigni, M: Agent based Ontological Mediation in IE Systems. In “Information Extraction in the Web Era: Natural Language Communication for Knowledge Acquisition and Intelligent Information Agents” LNAI 2700, Springer 2003
13. Garshol, L. G. and Moore, G.: Topic Maps — XML Syntax. (16 March 2004)
14. Dean, M. and Schreiber, G. editors.: OWL Web Ontology Language Guide. 2004. W3C Recommendation (10 February 2004).
15. De Bruijn, J. and Polleres, A.: Towards an Ontology Mapping Specification Language for the Semantic Web. DERI Technical Report (30 June 2004).