OWL-Based Approach for Semantic Interoperability

Seksun Suwanmanee, Djamal Benslimane
LIRIS Laboratory
University of Lyon 1
Villeurbanne, France
firstname.lastname@liris.cnrs.fr

Philippe Thiran
AIS Group, Computer Science Department
Eindhoven University of Technology
Eindhoven, the Netherlands
ph.thiran@tue.nl

Abstract

The number of web-based information systems has been increasing since Internet became the global open network accessible for all. The recent Semantic Web that provides supplementary meaningful information (meta-data) about Web resources facilitates automatic processing of machines and interoperability between different systems. In this paper, we focus on an integration of heterogeneous data sources in the Semantic Web context using a semantic mediation approach based on ontology. We use the ontology description language OWL to formalize ontologies of different resources and to describe their relations and correspondences allowing the semantic interoperability between them. We propose an architecture adopting mediator-wrapper approach for a mediator based on OWL. Some illustrations of semantic mediation using OWL are also presented in the paper.

1. Introduction

Since the "explosion" of internet, the technology in this area has been progressing and the data available on the Web have extraordinarily augmented, not only in quantity of information but also in quality and in various forms(simple HTML web pages, multi-media web pages, semi-structured and structured data resources, web services, etc). Many information systems can now expose their data via the internet that facilitates the public and remote access. The autonomous data sources that used to work locally become online accessible and ready to communicate with other systems. In our work, we concentrate on the integration of information systems in the internet. These autonomous information systems provide data in a common knowledge domain. Typically they are modeled and implemented differently according to thier local requirements. This may cause conflicts in several levels e.g. different platforms, schemas, data models. In order to construct the interoperability between them, an appropriate mediation system is necessary. Here we propose an approach based on a semantic mediation which uses ontology for representing the different data sources of a knowledge domain. The semantic mediation can play an important role in the context in which information may not be processed from only one data source, but instead from combinations of multiple heterogeneous data sources with different representations of a common domain.

Here we focus on heterogeneous information systems in the Semantic Web which allows meaningful descriptions and meta-data of the web resources. The meta-data of these data sources are necessary for the semantic mediation and also for other automatic applicaitions. That is why RDF/RDFS[1] was proposed by W3C[2] for the formulation of machine-readable meta-data in the Semantic Web. Then to support the needs of ontology language for the web resources, OWL[3]was crated as an extension of the RDF/RDFS founded on the semantics of description logics. Recently, OWL has become the determinant standardization effort of the international research community in this area. This implies that in the future we will see many ontologies in specific knowledge domains expressed in OWL. It is of crucial importance therefore to be able to integrate ontologies in order to provide the interoperability of different independent data sources.

The approach based on RDF and OWL ontology is relatively new in this research area and continue progressing. To cite a few examples, in [1] an RDF ontology-based approach is proposed for integrating different XML data sources. Furthurmore, an RDF-based architechture is presented in [14] in the context of semantic integration of heterogeneous data sources. A work based on OWL is also studied in [11] which focuses on schema mappings of XML data sources. Our main contributions in this paper are a gen-

1 http://www.w3.org/RDF/
2 http://www.w3.org
3 http://www.w3.org/2004/OWL/

eral architecture of OWL ontolgoy-based approach and a demonstration of using OWL language for a semantic interoperability of different data sources.

The rest of this paper is organized as follows. In the next section we describe main characteristics of OWL. Then we illustrate our problems with some motivating examples. In the next, we present the general architecture of our approach and then describe in detail the ontology mediation using OWL. Then some results of preliminary experiments will be explained. In the end we will conclude our current work and the future development.

2. OWL as ontology language

An ontology describes concepts and relations for representing and defining a specific knowledge domain. Essentially, it consists of a hierarchical description of concepts in a domain, along with descriptions of the properties of each concept and maybe instances of concepts.

As mentioned in many works such as [7],[13] ontologies can play an important role in the semantic mediation by providing a source of shared and precisely defined terms that can be used in meta-data.

RDF(Resource Description Framework), and RDF Schema(RDFS) were widely accepted as a formal language of meta-data describing any web resources. RDFS in particular is recognisable as an ontology knowledge representation language: it talks about classes and properties(binary relations), range and domain constraints(on properties),and subclass and subproperty (subsumption) relations. RDFS has, however, some limitations that cause difficulties for automated reasoning process. A new web ontology language such as DAML+OIL is developed based on the RDF model. It is known later as ancestor of OWL(Web Onotology Language) that is now officially recommended as ontology language for the Semantic Web by W3C.

OWL uses the same syntax as RDF (and RDFS) to represent ontologies. It may appear in several formats such as RDF/XML serialization, N-Triples, N3 and abstract syntax. In this paper OWL codes will be in RDF/XML format when we want to explicit the use of constructs and the abstract syntax format will be used for shorter codes.

Concretely, an OWL ontology consists of definitions and descriptions of concepts (or classes) and relations (or properties) between them. As an extension of RDF/RDFS, OWL uses some basic elements of RDF/RDFS such as `rdf:subClassOf`, `rdfs:domain`, etc. It also provides elements of the language which have specitifc semantics for defining classes, properties and describing their hierarchy and also their properties. The following paragraph explains briefly some fundamental elements of the language.

All classes in an OWL ontology are typed `owl:Class`. The expression `rdfs:subClassOf` decribes an inclusion relation between classes. `owl:equivalentClass` is used to declared the equivalence of classes. An enumerated class can be formed by the construct `owl:oneOf` which declares instances of a class. The properties are of two main types: `owl:DatatypeProperty` and `owl:ObjectProperty`. A *datatype property* is a binary relation that associates an individual of a class to a value(or values) of standard data types defined in accordance with XML Schema datatypes(`xsd`[4]) such as integer, string. On the other hand, an *object property* relates individuals of classes (or of a same class). When a property is defined, we may also specify its domain (`rdfs:domain`) and its range(`rdfs:range`). In addition, in OWL some specific characteristics such as transivitiy(`owl:TransitiveProperty`), symmetry (`owl:TransitiveProperty`), can be associated to a property.

OWL are classified into three dialects: OWL lite, OWL DL(description logic) and OWL Full. In this paper we are interested in OWL DL for it has enough expressivity and a decidable reasoning mechanism [8].

3. Running example

The following example ontologies are presented here to illustrate problems of semantic heterogeneity and the proposed solution. Suppose that there exists a school of music, say *SchoolA*, offers music classes for people who are interested in learning and practicing musical instruments. This school may have a local DBMS for the management of music classes, teachers and students.

The ontology representing this school concerns on some specific information about music classes and involved people. Some parts of the ontology of *SchoolA* are described as follows. There are 3 main concepts (or classes): *people*, *music class* and *music class level*. The class `People` includes people in the school that are classified into two categories: `Professor` and `Student`. `MusicClass` contains music classes which are classified by instrument (e.g. piano class, violin class, guitar class, etc). `MusicClassLevel` is an enumerated class containing three different levels of music classes: `Beginner`, `Intermediate` and `Advanced`. A class may possess some datatype properties and/or object properties. For example, the class `People` may have datatype properties such as name, age, sex, address, etc. The class `Professor` has an object property `teach` which associates individuals of Professor to an individual(or individuals) of `MusicClass`. In other words, `Professor` is the domain and `MusicClass` is the range of the property `teach`. Other details of the ontology are shown graphically in Figure 1. The OWL ontology corresponding to this ontology are described in OWL abstract syntax as follows.

```
Ontology ( SchoolA
  Class (MusicClass partial)
  Class (PianoClass partial MusicClass)
  Class (ViolinClass partial MusicClass)
  Class (MusicClassLevel complete
         oneOf(Beginner Intermediate Advanced) )
  Class (People partial)
```

4 http://www.w3.org/2001/XMLSchema#

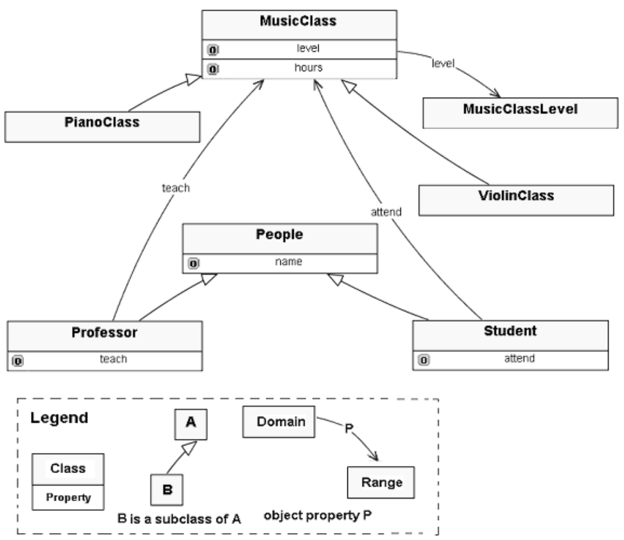

Figure 1. Ontology of *SchoolA*

```
  Class (Student partial People)
  Class (Professor partial People)
  ObjectProperty(attend
    domain(Student)   range(MusicClass))
  ObjectProperty(teach
    domain(Professor) range(MusicClass))
  ObjectProperty(level
    domain(MusicClass) range(MusicClassLevel))
  DatatypeProperty(name
    domain(People)  range(xsd:string))
  DatatypeProperty(hours
    domain(MusicClass) range(xsd:positiveInteger))
 )
```

Suppose that there is another school of music, say *SchoolB*, which models its information in a different way. This school uses its own vocabulary that may be semantically identical or different from other schools. The hierarchy of concepts is also designed differently. The schematic description of the ontology of *SchoolB* is displayed in Figure 2. This ontology contains the class `MusicCourse` which are divided into 3 subclasses by the level of the course: `BeginnerCourse`, `IntermediateCourse` and `AdvancedCourse`. Each course is characterized by properties such as `weeks` indicating the course duration, `useInstrument` indicating the instrument used in that course, `taughtBy` indicating the course instructor, etc. The rest of the ontology can be easily read from the diagram in Figure 2. The OWL ontology below described in abstract syntax format corresponds to the ontology of *SchoolB*.

```
Ontology ( SchoolB
  Class (MusicCourse partial)
  Class (BeginnerCourse partial MusicCourse)
  Class (IntermediateCourse partial MusicCourse)
  Class (AdvancedCourse partial MusicCourse)
  Class (Instrument complete
          oneOf(Piano Violin Cello) )
  Class (People partial)
  Class (Student partial People)
  Class (Instructor partial People)
  ObjectProperty(taugtBy
     domain(MusicCoruse)   range(Instructor))
```

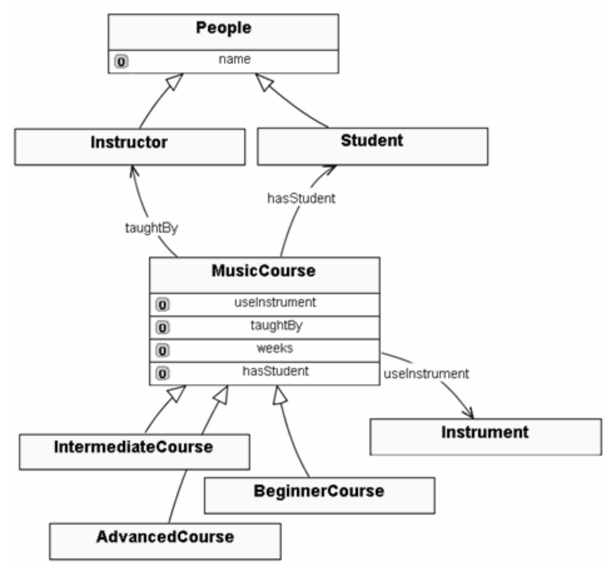

Figure 2. Ontology of *SchoolB*

```
  ObjectProperty(hasStudent
     domain(MusicCoruse) range(Student))
  ObjectProperty(useIntstrument
     domain(MusicCourse) range(Instument) )
  DatatypeProperty(name
     domain(People)     range(xsd:string))
  DatatypeProperty(weeks
     domain(MusicClass  range(xsd:positiveInteger))   )
```

It is obvious that the two example ontologies are different in several aspects. The different terms may be used for the same concepts such as Professor/Instructor, MusicClass/MusicCourse. This classical naming conflict can be solved by the equivalent class mappings. The music class classifications which use different criteria in two example ontologies raises more complicated problems. For example, there is no obvious corresponding class in *SchoolB* for `PianoClass` of *SchoolA*. The class mappings for this type of conflict requires specific restrictions on properties of concerned classes. Regarding properties, there are some similar ploblems such as naming conflict. The domain and range of properties may also cause some differences. For instance, in *SchoolA* the property `teach` relates a music teacher to a class (or classes) whereas the property `taughtBy` in *SchoolB* does the contrary. An appropriate mapping for properties are necessary to solve these kinds of problem.

In a general context we have a set of independent local data sources of a common domain and we need to share and exchange information among them. Each data source can be represented by its proper ontology that uses a certain vocabulary with specific semantics. An appropriate mediation system is needed for allowing the interoperability of different data sources. This mediation system must provide a means to overcome the semantic heterogeneity between the local systems and also a means to access to local data with transparency as much as possible. We will describe our proposed approach based on the formalization in OWL.

4. Approach

In this section, we present our approach based on ontology for semantic integration of heterogeneous data sources in the Semantic Web context.

4.1. Overview

In general view, our system consists of a collection of data sources and a mediator that facilitates the access to local data and reconciles semantic conflicts among those local systems. Our approach adopts a so-called mediator-wrapper architecture that allows local systems to function independently while the remote access can be done via a mediator and adaptable wrappers. This mediation system provides a transparent access of different local sources to the user. Illustrated in Figure 3, the architecture of our system may be divided into three layers:

- Source layer contains a set of autonomous sources of structured data such as database systems (relational or object), Web documents, XML repositories. Each local system models data of a common domain in its own way according to its requirements and application.
- Wrapper layer contains wrappers for each local resource. Those wrappers allow the communication between local systems and the mediator. In the context of Semantic Web, the wrapper provides an OWL ontology representing a data source and a means to access and to query the local source. We will discuss about the wrapper in the next subsection.
- Mediating layer contains a mediator which allows the interoperability of the local data sources. One of its main functions is to integrate local ontologies in order to guarantee a global access of local sources. Since different local ontologies may present some semantic conflicts, ontology mappings are necessary to overcome these differences. The mediator also contains a reasoning engine that works on the ontologies and the mappings and a query processor which allows the users to have a global and transparent access of all local sources. For this paper, the query processing is beyond the scope. Here we focus on the ontology mediation. In the following subsection we will show how we can formulize ontology mapping in OWL.

4.2. Wrappers

Wrappers are developed on top of each local data sources and provide a standard and common interface to facilitate and homogenize their access.

This interface is made up of: (1) a local ontology of the wrapped data source, expressed in OWL and (2) a query language which uses the semantics defined in the local ontology.

Local Ontology
In order to export the local sources in OWL, we need to define how a source schema expressed in any modelling language can be mapped to the OWL data model. For this purpose existing works can be used. Many relational schemas to XML schemas translations have been proposed such as the work in [10]. In citeLehti04:SAINT a mapping of XML schema to OWL is presented.

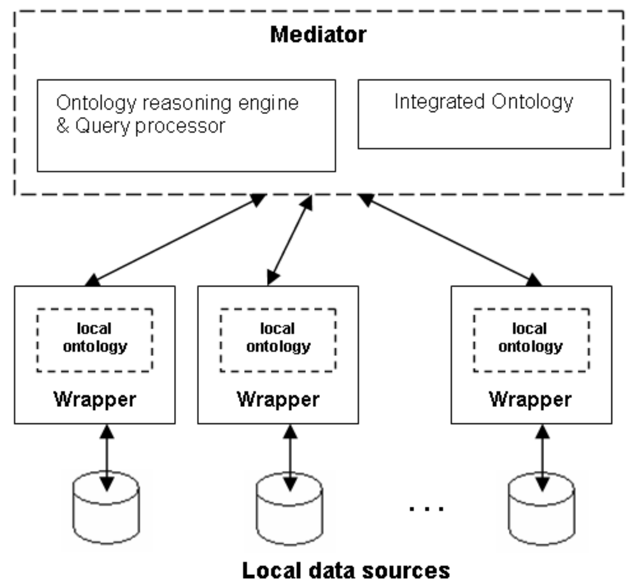

Figure 3. General architecture

There is also a more concrete tool such as D2R[5] which contributes a flexible mapping language to transform relational data into RDF graph model that will be easily adapted to OWL format.

Query Language
In our case, queries need to be based on OWL; that means that the query language needs a formally defined semantics for the OWL data model. Therefore one could use and slightly modify OQL or one of the RDF query languages ([2], [9]) because there are also defined on a graph models. Recently, [11] proposed the query language SWQL which specializes in OWL.

4.3. Ontology mediation

The ontology mediation in our approach consists of mappings of elements of different OWL ontologies. OWL provides sufficient elements for expressing relations between classes and between properties as well and these expressions are not limited in a same ontology. As a result, we can apply OWL to describe the mappings of different ontologies. Our objective is to obtain an integrated ontology which contains semantic mappings of different local ontologies.

We illustrate how to use OWL as ontology mapping language by showing the mapping between previous motivating examples.

Ontology Importing
First of all, it is important to specify the predefined involved ontologies by their URI so that the rest of ontology description will be able to refer to the existing elements that are previously defined in a local ontology. This reference is described by the OWL expression `owl:import`. Here is an example of OWL code of importing our predefined ontology *SchoolA* and *SchoolB*.

```
<owl:Ontology rdf:about="">
 <rdfs:label>Integration of SchoolA and SchoolB
```

5 http://www.wiwiss.fuerlin.de/suhl/bizer/d2rmap/D2Rmap.htm

```
 </rdfs:label>
 <owl:imports
   rdf:resource="http://music.school/schoolA"/>
 <owl:imports
   rdf:resource="http://music.school/schoolB"/>
</owl:Ontology>
```

Class Mappings

Basically, inclusion, equivalence and disjunction relations between classes allow us to describe a hierarchical structure of classes in an ontology. However we can also apply these class relations to establish mappings between classes from different ontologies. One class of an ontology may be considered as a subclass of another class of another ontology. They can also be equivalent(`owl:equivalentClass`). On the other hand, two classes which have no individual in common may be explicitely declared disjoint(`owl:disjointWith`) with each other.

Here are examples of simple class mappings which describe some equivalent concepts in the ontologies *SchoolA* and *SchoolB* such as `MusicClass/MusicCourse`, `Professor/Instructor`, etc.

```
<owl:Class rdf:about="&schoolA;MusicClass">
   <owl:equivalentClass
          rdf:resource="&schoolB;MusicCourse"/>
</owl:Class>
<owl:Class rdf:about="&schoolA;People">
    <owl:equivalentClass
           rdf:resource="&schoolB;People"/>
</owl:Class>
<owl:Class rdf:about="&schoolA;Professor">
    <owl:equivalentClass
          rdf:resource="&schoolB;Instructor"/>
</owl:Class>
```

OWL provides some expressions to construct a concept that represents a class of individuals which satisfy some common conditions. A complex class can also be formed by classical set operations like union, intersection and complement. The restrictions and complex class constructions allow us to describe complicated and precise classes. In OWL we can specify a restriction on certain property according to its associated value with `owl:hasValue)`, its range of values(existential condition with `owl:someValuesFrom` or universal condition with `owl:allValuesFrom`) and its cardinality(with `owl:min/maxCardinality` or `owl:cardinality`).

These following examples demonstrate some more complex class mappings.

The class `BeginnerCourse` of *SchoolB* can be considered as any music classes of *SchoolA* which have the value `Beginner` in its property `level`. This mapping can be expressed in OWL as follows.

```
<owl:Class rdf:about="&schoolB;BeginnerCourse">
   <owl:equivalentClass>
     <owl:Restriction>
      <owl:hasValue
          rdf:resource="&schoolA;Beginner"/>
      <owl:onProperty
          rdf:resource="&schoolA;level"/>
     </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

The mapping for `PianoClass` of *SchoolA* to *SchoolB* can be expressed in similar way by restricting the appropriate value of property `useInstrument`. The mapping for `ViolinClass` is more complicated if we consider any music courses of *SchoolB* using either violin or cello as a violin class of *SchoolA*. The union operation is required here for the mapping that are described below.

```
<owl:Class rdf:about="&schoolA;ViolinClass">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
       <owl:Restriction>
        <owl:onProperty
          rdf:resource="&schoolB;useInstrument"/>
        <owl:hasValue
          rdf:resource="&schoolB;Violin"/>
       </owl:Restriction>
       <owl:Restriction>
        <owl:onProperty
          rdf:resource="&schoolB;useInstrument"/>
        <owl:hasValue
          rdf:resource="&schoolB;Cello"/>
       </owl:Restriction>
      </owl:unionOf> </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

Property Mappings

We determine a relation between two properties by comparing their members. Three possible relations of properties are inclusion, equivalence and inverse. *P1* is a subproperty (`rdfs:subPropertyOf`) of *P2* means that if *P1(x,y)* holds then *P2(x,y)* holds. *P1* and *P2* are equivalent properties (`owl:equivalentProperty`) when *P1(x,y)* if and only if *P2(x,y)*. *P1* is an inverse (`owl:inverseOf`) of *P2* when *P1(x,y)* if and only if *P2(y,x)*. Some examples of property mappings are presented below. In our running examples, we see that the properties `name` of *SchoolA* and `name` of *SchoolB* which indicate the name of a person, have the same meaning. They are equivalent properties. On the other hand, the property `teach` of *SchoolA* is exactly the reverse of `taughtBy` of *SchoolB*. We can also map the property attend as the inverse property of `hasStudent`. These mappings are expressed in OWL as follows.

```
<owl:DatatypeProperty rdf:about="&schoolA;name">
  <owl:equivalentProperty
      rdf:resource="&schoolB;name"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:about="&schoolA;teach">
  <owl:inverseOf
       rdf:resource="&schoolB;taughtBy"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&schoolA;attend">
  <owl:inverseOf
      rdf:resource="&schoolB;hasStudent"/>
</owl:ObjectProperty>
```

The ontology mappings described above will allow a reasoning engine to build an integrated ontology with global hierarchy of all classes from local ontologies and to detect a consistency of them. The mapping rules are also useful for query processing and query reformulations.

5. Preliminary experiments

As we are in an early step of our work, we carry out some experiments on our ontology mediation using existing OWL tools. To develop an OWL ontology, a simple text editor or XML editor may be sufficient for coding an OWL ontology in XML/RDF serialization syntax.

Racer system [4] is one of description logic reasoners publicly available. OWL compatibility is a new feature of Racer that allows user to use it as an ontology reasoning engine. We can load an OWL ontology into Racer system using the particular interface program called RICE and then it will verify a consistency of the ontology and display a general classification of all concepts defined in the ontology. By loading an integrated ontology containing ontology importations and description of ontology mappings, RICE program shows us a global hierarchy of all classes from different ontologies. We can select a particular class and see all instances of the underlying class. Besides, RICE provides an interactive querying system that allows us to make queries over loaded ontologies to the running Racer server. However these queries are formed in the Racer syntax.

We can formulate our query in OWL by a class definition in the integrated ontology. A query can be described by using any terms of imported ontologies and the result of the query comes from all concerning local ontologies. Here are some examples of query that we formulate in OWL.

```
<owl:Class rdf:ID="Q_advancedPianoClass">
 <owl:intersectionOf rdf:parseType="Collection">
   <owl:Class rdf:about="&schoolA;PianoClass"/>
   <owl:Classrdf:about="&schoolB;AdvancedCourse"/>
 </owl:intersectionOf>
</owl:Class>
```

This new class includes all music classes of *SchoolA* and *SchoolB* which are contained in the intersection of `PianoClass` and `AdvancedClass`. It presents all advanced piano courses of the two ontologies.

6. Conclusion and future work

In this paper, we propose an approach based on OWL for semantic integration of heterogeneous data sources in the context of Semantic Web. We described our mediator-wrapper architecture and the ontology mediation with OWL. Then we showed some experiments on OWL ontology integration.

Our current work does not yet contribute a real functional system but it seems to be promising and the development of components mentioned in our approach (e.g. wrapper for OWL, adaptable OWL query language) are in progress and there are many active research works in this area.

Many open issues are not discussed in this paper such as instances in ontology, query processing, to mention a few. These are subjects for future work.

It is also interesting that we move toward an open distributed system which is suitable for the Web context, especially, peer-to-peer architecture. There are several approaches([12],[5], [6]) proposed in the literature for this particular system. Distributed ontology is also studied in [3]. We can effectively improve our approach to be more flexible and dynamic for a distributed context. This is a part of perspectives of our work.

References

[1] An ontology-based framework for XML semantic integration. In *Proc. of IDEAS 2004*. IEEE Computer Society, 2004.

[2] F. Fransincar, G. J. Houben, R. Vdovjak, and P. Barna. RAL: An algebra for querying RDF. *World Wide Web, Internet and Web Information Systems(WWW)*, 7(1):83–109, 2004.

[3] F. Goasdoué and M.-C. Rousset. Querying distributed data through distributed ontologies : a simple but scalable approach. *IEEE Intelligent Systems*, 18(5):60–65, 2003.

[4] V. Haarslev and R. Möller. Racer system description. In *Proc. of Int. Joint Conference on Automated Reasoning(IJCAR'2001)*. Springer-Verlag, June 2001.

[5] A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *Proc. of the Int. Conference on Data Engineering(ICDE)*, 2003.

[6] A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov. Peer data management systems: Infrastructure for the semantic web. In *Proc. of WWW Conference*, 2003.

[7] I. Horrocks and P. Patel-Schneider. Three theses of representation in the semantic web, 2003.

[8] I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfability. In *Proc. of the 2003 Int. Workshop on Description Logics (DL2003)*, September 2003.

[9] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A declarative query language for RDF. In *In The 11th Intl. World Wide Web Conference (WWW2002)*, 2002.

[10] D. Lee, M. Mani, F. Chiu, and W. W. Chu. Net & cot: translating relational schemas to XML schemas using semantic constraints. In *CIKM*, pages 282–291, 2002.

[11] P. Lehti and P. Fankhauser. XML data integration with OWL: Experiences and challenges. In *Proc. of Symposium on Applications and the Internet (SAINT 2004)*, pages 160–170, 2004.

[12] A. Löser, W. Siberski, M. Wolpers, and W. Nejdl. Information integration in schema-based peer-to-peer networks. In *Proc. of CAiSE*, pages 258–272, 2003.

[13] E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[14] R. Vdovjak and G.-J. Houben. RDF-based architecture for semantic integration of heterogeneous information sources. In *Workshop on Information Integration on the Web*, pages 51–57, 2001.