OWL based XML Data Integration

Manjula Shenoy K
Manipal University
CSE department
MIT Manipal, India

K.C.Shet, PhD.
N.I.T.K.
CSE department, Suratkal
Karnataka, India

U. Dinesh Acharya, PhD.
ManipalUniversity
CSE department
MIT, Manipal, India

ABSTRACT
Data integration helps in manipulating data transparently across multiple distributed databases. The purpose of integration system is to provide a unified global view to the user over various heterogeneous data sources. To answer user queries, a data integration system employs a set of semantic mapping between global and local schema. Such integration has challenges of creation of local and global schema and mappings generation. This paper proposes an approach to address such a data integration system for data sources having heterogeneity in structure and semantics. Here data sources chosen are assumed to be in XML format with different schema

General Terms
Data integration, ontology, information integration. Artificial Intelligence.

Keywords
OWL, XML, XML Schema, Schema mappings, ontology mapping.

1. INTRODUCTION
Data integration helps in manipulating data transparently across multiple distributed databases. Independent data sources are often heterogeneous in nature. This heterogeneity is of three type, namely Syntactic, Schematic and Semantic heterogeneity. Syntactic heterogeneity comes from usage of different languages for modeling data sources. Schematic heterogeneity results from different structures of source schemas and semantic heterogeneity arises when different sources contain instances with different meanings or interpretations of data in various contexts. To achieve data interoperability, the issues posed by data heterogeneity need to be eliminated. The key notation introduced by Semantic Web, so called ontology can be used to solve the problem of data integration. Ontology is defined as formal and explicit specification of a shared conceptualization. That is it defines a particular domain in terms of concepts, relations, properties, axioms in machine readable form. OWL is the standard language to express this ontology.

2. RELATED WORK
Data integration is relevant to a number of applications including enterprise information integration, medical information management, geographical information systems, and e-Commerce applications. Based on the architecture, there are two different kinds of systems: central data integration systems (3) and peer-to-peer data integration systems (3). A central data integration system usually has a global schema, which provides the user with a uniform interface to access information stored in the data sources. In contrast, in a peer-to-peer data integration system, there are no global points of control on the data sources (or peers). Instead, any peer can accept user queries for the information distributed in the whole system. The two most important approaches for building a data integration system are Global-as-View (GaV) and Local-as-View (LaV) (7). In the GaV approach, every entity in the global schema is associated with a view over the source local schema. Therefore querying strategies are simple, but the evolution of the local source schemas is not easily supported. On the contrary, the LaV approach permits changes to source schemas without affecting the global schema, since the local schemas are defined as views over the global schema, but query processing can be complex.

The advent of XML has created a syntactic platform for Web data standardization and exchange. However, schematic data heterogeneity may persist, depending on the XML schemas used (e.g., nesting hierarchies). Likewise, semantic heterogeneity may persist even if both syntactic and schematic heterogeneities do not occur (e.g., naming concepts differently). The key notation introduced by Semantic Web, so called ontology can be used to solve the problem of data integration. Ontology is defined as formal and explicit specification of a shared conceptualization. In this definition, “conceptualization" refers to an abstract model of some domain knowledge in the world that identifies that domain's relevant concepts. “Shared" indicates that ontology captures consensual knowledge, that is, it is accepted by a group. “Explicit" means that the type of concepts in ontology and the constraints on these concepts are explicitly defined. Finally, “formal" means that the ontology should be machine understandable. Ontologies were developed by the Artificial Intelligence community to facilitate knowledge sharing and reuse (3). Carrying semantics for particular domains, ontologies are largely used for representing domain knowledge. A common use of ontologies is data standardization and conceptualization via a formal machine-understandable ontology language. Existing ontology languages include OWL, RDFS, DAML+OIL and so on. RDFS (RDF Schema) is a language for describing vocabularies of RDF data in terms of primitives such as rdfs:Class, rdf:Property, rdfs:domain, and rdfs:range. In other words, RDFS is used to define the semantic relationships between properties and resources. DAML+OIL (DARPA Agent Markup Language-Ontology Interface Language) is a full-fledged Web-based ontology language developed on top of RDFS. It features an XML-based syntax and a layered architecture. DAML+OIL provides modeling primitives commonly used in frame-based approaches to ontology engineering, and formal semantics and reasoning support found in description logic approaches. It also integrates XMLSchema data types for semantic interoperability in XML. OWL (Web Ontology Language) is a semantic markup language for publishing and sharing ontologies on the Web. It is developed as a vocabulary extension of RDF and is derived from DAML+OIL. Ontologies have been extensively used in data integration systems because they provide an explicit and machine-understandable conceptualization of a domain. They have been used in one of the three following ways: *Single ontology approach*, here all source schemas are directly related to a shared global ontology that provides a uniform

interface to the user (3). However, this approach requires that all sources have nearly the same view on a domain, with the same level of granularity. A typical example of a system using this approach is SIMS (3). *Multiple ontology approach* here each data source is described by its own (local) ontology separately. Instead of using a common ontology, local ontologies are mapped to each other. For this purpose, additional representation formalism is necessary for defining the inter-ontology mappings. The OBSERVER system (3) is an example of this approach. *Hybrid ontology approach is a* combination of the two preceding approaches. First, a local ontology is built for each source schema, which, however, is not mapped to other local ontologies, but to a global shared ontology. New sources can be easily added with no need for modifying existing mappings. Our layered framework (3) is an example of this approach. The single and hybrid approaches are appropriate for building central data integration systems, the former being more appropriate for GaV systems and the latter for LaV systems. Here we have proposed an OWL based data integration system for Data sources in XML format having heterogeneity in Schema and Semantics.

3. PROPOSED SYSTEM

The proposed system is as shown in Fig.1. Here two data sources in XML format are defined. The schema of the data sources is first converted to local ontology in OWL together with the mapping table. A global ontology for these two local ontologies is considered and using suitable mapping method, mapping tables for local to global or global to local are generated. User can query the system using global ontology and this query is rewritten using mapping tables in XQuery format to query the actual sources and the result of these is merged and presented to the user.

3.1 Input to the system

Here we define the two data sources taken in XML format having schematic and semantic heterogeneity. Fig. 2. lists the first data source and Fig. 3. Lists its schema. Fig. 4. Lists second data source and its associated schema in Fig.5.

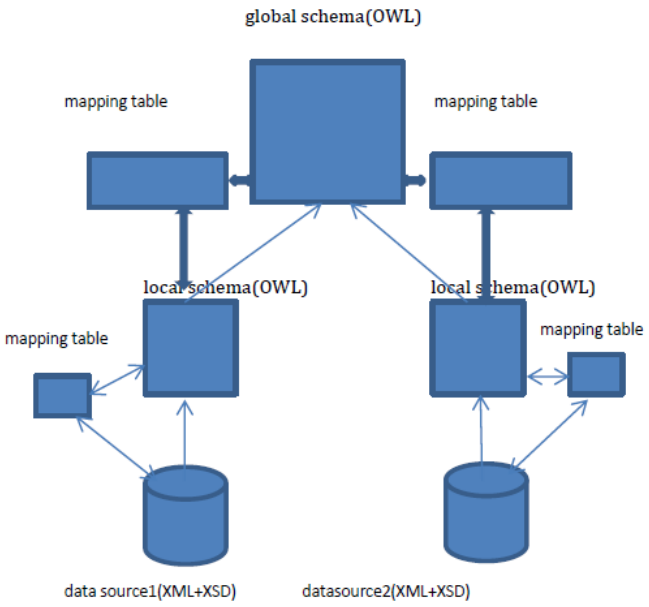

Fig.1.Proposed System Architecture

```
<department>
        <csdepartment>
        <faculty>  <name>MMRao</name>
                <pub>p01</pub>
                <pub>p02</pub>
        </faculty>
        <faculty> <name>MJRao</name>
                <pub>p03</pub>
                <pub>p02</pub>
        </faculty>
        <publication>       <id> p01</id>
                            <title> t1</title>
                            <type>book</type>
        </publication>
        <publication>        <id> p03</id>
                            <title> t3</title>
                            <type>Journal</type>
        </publication>
        <publication>       <id> p02</id>
                            <title> t2</title>
                            <type>conference</type>
        </publication>
        </csdepartment>
        <ecdepartment>
         <faculty>
                            <name>MRao</name>
                            <pub>p06</pub>
                            <pub>p05</pub>
        </faculty>
        <faculty>
                            <name>MJRao</name>
                            <pub>p04</pub>
                            <pub>p05</pub>
        </faculty>
        <publication>        <id> p04</id>
                            <title> t4</title>
                            <type>book</type>
```

```
        </publication>
        <publication>      <id> p05</id>
                           <title> t5</title>
                           <type>Journal</type>
        </publication>
        <publication>       <id> p06</id>
                           <title> t6</title>
                           <type>conference</type>
        </publication>     </ecdepartment>
</department>
```

Fig. 2. First Data Source in XML Format

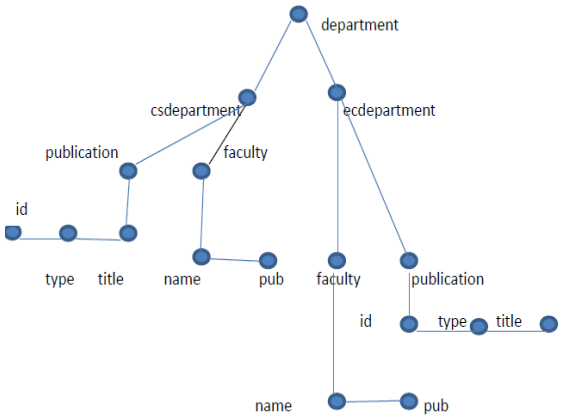

Fig. 3.Schema of first data source

```
<publications>
        <publication>
        <pubid>p01</pubid>
        <authors>xxy</authors>
        <authors>xxx</authors>
        <title>t1</title>
        <type>book</type>
</publication>
<publication>
        <pubid>p07</pubid>
        <authors>yy</authors>
        <authors>xxyy</authors>
        <title>t1</title>
        <type>book</type>
        </publication>
<author>
    <name> xxy </name>
    <dept> cs</dept>
</author>
<author>
        <name> xxx </name>
        <dept> cs</dept>
</author>
<author>
    <name> yy </name>
    <dept> ec</dept>
</author>
<author>
        <name> xxyy </name>
    <dept> ec</dept>
</author>
</publications>
```

Fig. 4. Second Data Source in XML Format

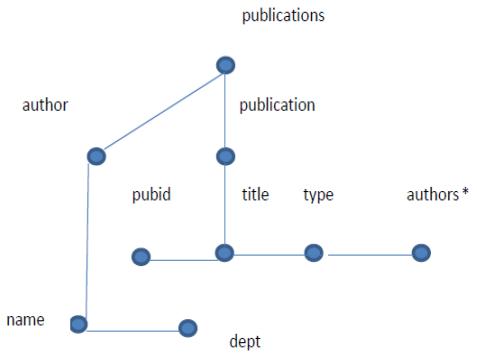

Fig. 5.Schema of second data source

3.2 Local ontologies designed

The Fig. 6. and Fig. 7 denote local ontologies designed for the schema for data sources. Fig.8 shows global ontology for these datasources.

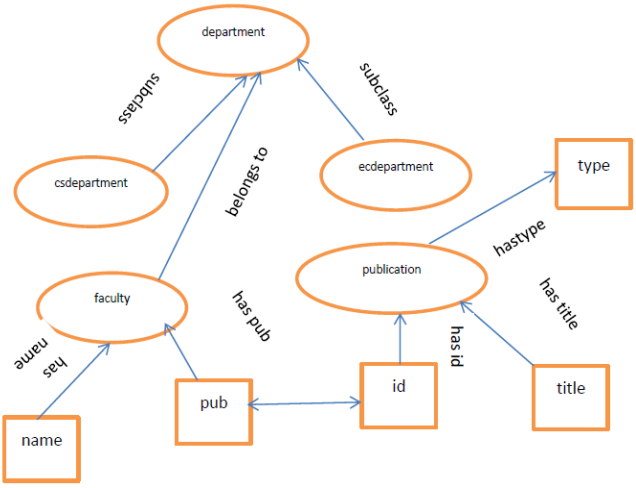

Fig. 6. Local Ontology for Data Source1

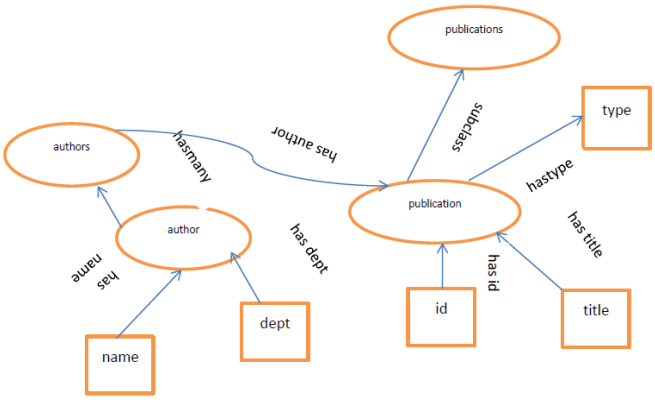

Fig. 7.Local Ontology for Data Source2

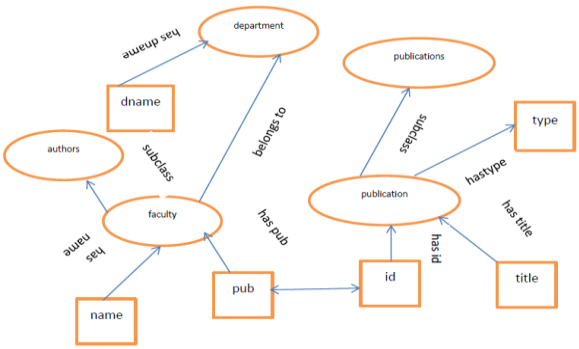

Fig. 8. Global ontology for data sources

3.3 Mapping of XML schema to local ontology

The following rules are used in mapping XML schema to local ontology.

1) The complex type is mapped to a concept.
2)Simple types are mapped to properties
3)Attributes are mapped to properties

The mapping generated so are stored in a local mapping table.

3.4 Mapping of local ontologies to global ontology

This mapping is done using an simple ontology mapping system developed by the authors. The output of mapping is stored as global to datasource1 and global to datasource2 mapping tables. These mapping are given in Table1 and Table 2.

Table 1. Mapping table for data source1

CONCEPTS			
	department	=	department
	csdepartment	≤	department
	ecdepartment	≤	department
	faculty	=	faculty
	faculty	≤	authors
	publication	≤	publications
	publication	≤	publication
PROPERTIES	hasname	=	hasname
	haspub	=	haspub
	hasid	=	hasid
	hastitle	=	hastitle
	hastype	=	hastype

Table 2. Mapping table for data source2

CONCEPTS			
	dept	≤	department
	authors	=	authors
	faculty	=	author
	publications	=	publications
	publication	≤	publication
PROPERTIES	hasdept	=	hasdname
	haspub	=	haspub
	hasid	=	hasid
	hastitle	=	hastitle
	hastype	=	hastype

3.5 Querying the integrated system

When the user poses a query q on the global ontology,the system rewrites q into the union q' of sub queries, one for each XML source. The sub queries are then executed over the XML sources to get the answers, which are then integrated (by using union) to produce the answer to q. Query rewriting in both directions is based on the mapping information

contained in the mapping table. Suppose the query posed towards global ontology is *list all faculty* expressed as

Select ?name, where

{ ?facutly rdf:type faculty.

?faculty ?name}

This query should be rewritten for data source1 as

doc("ds1.xml")//faculty/name

and for data source2 as

doc("ds2.xml")//author/name

3.6 Results

The system was tested using Sedna XML data base and Xquery processor and a few XML files which have different type of heterogeneity. The accuracy of querying result was based upon accuracy of the mapping system used. The accuracy for different set of files is listed in Table 3.

Table 3. Result Analysis

Sl No of Sets	Accuracy achieved
1	60%
2	65%
3	63%
4	66%

4. CONCLUSION

The paper has explained a small prototype to address the problem of data integration. It has taken an example of XML data sources which are heterogeneous in nature and developed a system to query it by considering it as a whole XML data by making the individual sources transparent. Future work is coming up with proper syntax for querying global ontology, and an algorithm to rewrite queries based on mapping tables.

5. REFERENCES

[1] J.Banerjee,W.Kim,H.-J.Kim, and H.Korth Semantics and implementation of schema evolution in Object-OrientedDatabases.In SIGMOD1987.

[2] I.Madhavan and A Halevy . Composing mapping among data sources.In VVLDB,2003.

[3] Huiyong Xiao. Query processing for heterogeneous dataintegration using ontologies.PhdThesis,2006.

[4] Chong-Shan Ran,Ma-Chuan Wang. An XML Schema based data integartion.In IEEE-2010

[5] Xiong Fengguang,Han Xie,KuangLiqun.Research and implementation of heterogeneous data integration based on XML.IEEE-2009

[6] A.Rajesh,S.K.Srivatsa. XML Schema Matching-using structural information.In International Journal of Computer Applications.2010.

[7] PatrickLehti,PeterFankhauser.XML data integration with OWL : Experiences and Challenges.IEEE-2004.

[8] Berners-Lee,T.,J.Hendler, et.al. "The Semantic web",ScientificAmerican 284(5),2001.

[9] Grigoris Antoniou,Frank Van Harmelen," Semantic web Primer" The MIT Press 2004.