Semantic Mappings for the Integration of XML and RDF Sources

Huiyong Xiao   Isabel F. Cruz   Feihong Hsu

Department of Computer Science
University of Illinois at Chicago, USA
{hxiao, ifc, fhsu}@cs.uic.edu

Abstract. A huge amount of data on the Web may be heterogeneous with respect to syntax, schemata and semantics. For instance, XML and RDF provide two completely different paradigms for modeling Web data. In this paper, we focus on the issue of mapping representations in an ontology-based framework that aims at integrating XML and RDF sources. We propose a solution that utilizes a new mapping language called RDF Mapping Schema, which is a meta-schema defined on top of RDF Schema.

1 Introduction

One of the primary obstacles in the Web information integration applications is the heterogeneity of the distributed data sources. These heterogeneities can be classified as *syntactic*, *schematic*, and *semantic* [5]. For instance, XML and RDF provide two completely different paradigms for representing Web data. There are currently many attempts to use a conceptual-level schema (ontology) [1, 2, 9, 10] or a conceptual-level query language [7, 8] to integrate heterogeneous data sources. As in our previous work [10], we propose an ontology-based approach to semantically integrate XML and RDF sources. The approach uses a mediating global ontology that is modeled in RDF and constructed using the global-as-view (GAV) approach [6]. Mappings are established between the global ontology and the RDF or XML sources, which then interoperate through these mappings by using a query translation mechanism.

In the aforementioned systems, it is commonly assumed that only one-to-one mappings are considered and that the only semantics of a mapping is *equivalence*. Unfortunately, this assumption results in the weak expressiveness for the mappings, and, furthermore, affects the correctness and accuracy of query answering for the system. Let us illustrate the mapping problem by using a concrete example shown in Figure 1, which lists three schemas of different data sources: two XML schemas defined by DTD files (represented as indented text) and an RDF schema represented as a graph. In particular, the limitation of one-to-one mono-semantic mappings leads to the following three shortcomings.

- The **degree of equivalence** cannot be distinguished, since a pair of mapped equivalent concepts may vary much in semantics. For instance, the mapping

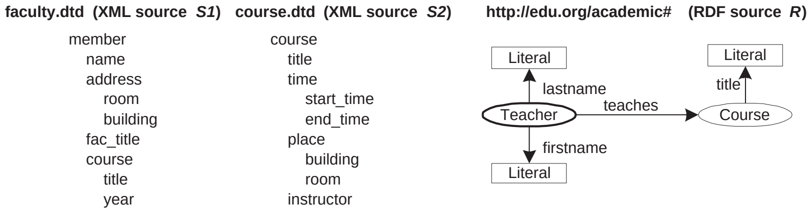

Fig. 1. An example of heterogeneous XML and RDF data sources.

process will take `instructor` in `course.dtd` as being equivalent to `member` in `faculty.dtd`, although actually `instructor` may be more general than `member`. In fact, the instances of `instructor` may contain a person who is not a faculty member (e.g., a teaching assistant).

– The semantic relationship involving a **one-to-many mapping** cannot be represented. For instance, `name` in `faculty.dtd` is semantically equivalent to the two concepts `lastname` and `firstname` in the RDF schema R.

– **Instance-level mappings**, which are required for the process of query answering, cannot be represented. For example, the instance of `name` in `faculty.dtd` may be mapped to R as a concatenation of `lastname` and `firstname`, whereas `instructor` in `course.dtd` is mapped to R as a concatenation of `firstname` and `lastname`.

In this paper, we address these issues and propose an RDF-based mapping language, which contributes to a more powerful representation of the mappings and then facilitates a more accurate query answering mechanism for the system. More specifically, a new RDF meta-schema (i.e., an RDF schema used to define RDF schemas), called RDF Mapping Schema (RDFMS), is defined to represent the semantic mappings. Our approach makes use of both the semantic mappings and the semantic query translation between the global ontology and the local data sources (represented in either XML or RDF), This approach enables semantic interoperation between the data sources by hiding their heterogeneities.

There currently exists the following approaches to the mapping representation problem in a data integration system. The Piazza system [11] enables integration of XML and RDF sources; it also uses a declarative XQuery-based mapping language ~~is~~ proposed for mapping representation. The mappings between RDF schemas and XML schemas are actually defined at a syntactic level, through mappings between XML and RDF/XML. In comparison, our improved framework uses RDFMS to define the mappings at a semantic level. The Clio project [12] represents the mappings using query (view) definitions. This facilitates the generation of the consistent data translations from source schema to target schema. However, the system only supports XML schemas and relational databases, and not RDF sources. The MOMIS system [3] supports semantic mappings between sources by using a common thesaurus, which contains three basic semantic relationships: synonym, hypernym (broader term), and hylonym

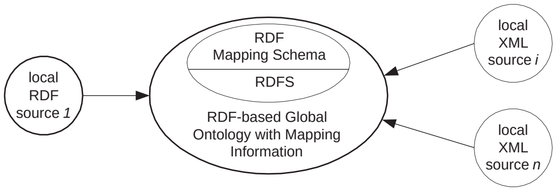

Fig. 2. The improved architecture of the integration framework.

(related term). In comparison, the RDFMS language provides richer semantics and more powerful expressiveness for the mapping representation.

In summary, our approach has the following advantages: (1) The RDF-based mapping information represented by the RDFMS language can be seamlessly combined with the RDF-based global ontology. Together, they form a network of semantic relationships of concepts, acting as the mediator for query processing. (2) The RDFMS mapping language is extensible so that users can define new ad-hoc mapping types based on the core of RDFMS. (3) The combination of an inferencing mechanism with RDFMS, to facilitate an automatic (or semiautomatic) mapping process and query translation, seems promising and reasonable.

The paper is organized as follows. Section 2 describes the architecture that we use to integrate XML and RDF sources. In Section 3, we discuss in detail the semantic mappings in our architecture and the mapping language that is used to represent them. We draw conclusions and discuss future work in Section 4.

2 Architecture

Figure 2 shows the architecture of our proposed semantic integration system. We use RDFS to model the global ontology, which is generated from the local XML and RDF sources and connects with them through semantic mappings. To solve the problems resulting from one-to-one mono-semantic mappings, we define a new RDF meta-schema, RDFMS, to represent the mappings. The operation of the system consists of the following two aspects: mapping and query processing.

Mapping. In the design time of the system, a local source gets connected to the global ontology by mapping its local schema to the global ontology through a process of *schema matching.* [1] In the meantime, the global ontology is constructed or enriched. We represent an RDF schema as a labeled digraph, called *RDF schema graph.* We also represent an XML schema using a labeled tree, called *XML schema tree*, with the labeled vertices being elements and attributes in the XML schema definition (e.g., DTD or XML Schema). Matching a local

[1] Schema matching is a basic problem in many database application domains. A taxonomy covering most of the existing approaches to schema matching has been devised [13]. Currently schema matching is usually performed manually or semiautomatically by utilizing some lexicon or thesaurus [3].

schema with the global ontology is essentially a process of matching an RDF schema graph or an XML schema tree with the global RDF ontology graph.

Query processing. During running time, the user may pose a query either over the global ontology or over any local RDF or XML source. Hence the framework supports two query processing modes: the *data-integration mode*, in which the query posed on the global ontology is reformulated into multiple subqueries over the XML and RDF sources (one subquery for each source), and the *peer-to-peer mode*, in which the query posed over one local source is reformulated into a query over other data sources. More details on query processing can be found in our previous work [10].

3 Semantic Mappings

3.1 Mapping Types

In this section, we discuss three types of semantic mappings based on examples, rather than giving a complete formal description.

Derivation mappings In our framework, *derivation mappings* can be further divided into the three classes: (1) SUPER: A SUPER B means that A is a more generalized concept of B. (2) SUB: A SUB B means that A is a more specialized concept of B. (3) EQU: A EQU B simply means A and B are semantically equivalent.

Operation mappings An *operation mapping* maps two or more concepts through operators, which define the transformation rules of the instances of the mapped concepts. An operation mapping is usually used together with a derivation mapping or a *constraint mapping* (see later discussion) to define a complete mapping. For instance, `name` in S_1 is equivalent to the combination of `firstname` and `lastname` in R. Thus we may denote this mapping as: `name` EQU (`firstname` AND `lastname`), where AND is the operator concatenating the string instances of `firstname` and `lastname`.

In essence, the interpretation of an operator defines the rules of instance transformation and is determined at runtime. For example, given A AND B, the AND operation may be interpreted as a sum of two numbers, a concatenation of two strings, a composition of two relations, or an intersection of two sets, etc. Which interpretation is correct, depends on the types of instances of A and B at runtime.

Constraint mappings *Constraint mappings* use constraint operators to specify the constraints that the instances of the mapped concepts must conform to. Based on this feature, constraint mappings may have three roles in the framework: (1) limitations for filtering the instances for query answering, (2) requirements of data consistency for data coordination, and (3) enabling queries with the constraints involving attributes from multiple local sources. Typical constraint operators include *comparison* (e.g., $\geq$, $=$, $\leq$, and $\neq$) and *disjoint* of two concepts. For example, for S_2 we may specify `start_time` $\leq$ `end_time`.

3.2 RDF Mapping Schema

Figure 3 shows a fragment of the definition of RDFMS in the W3C N3 language [4]. We define the three derivation mapping operators (SUPER, SUB, and EQU) respectively as three RDF properties (`super`, `sub`, and `equ`) which individually connects two mapped concepts. Likewise, the constraint mapping operators (e.g., disjoint) are also defined as RDF properties. They can be recursively applied to define a constraint involving more than two RDF resources. We particularly define the operators (e.g., AND) in operation mappings as RDF classes, which inherit the super class `Operator`. The RDF property `operand` is used to connect an operator and an operand (an RDF Resource). To connect multiple operands to a single operator, subproperties (e.g., `operand_1` and `operand_2`) can be defined inheriting `operand`.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfms: <http://www.cs.uic.edu/ advis/rdfms#> .
rdfms:super rdf:type rdfs:Property ;
             rdfs:domain rdfs:Class ;
             rdfs:range rdfs:Class .
rdfms:Operator rdf:type rdfs:Class .
rdfms:And rdf:subClassOf rdfms:Operator .
rdfms:operand rdfs:type rdfs:Property ;
               rdfs:domain rdfms:Operator ;
               rdfs:range rdfs:Resource .
# ...
rdfms:disjoint rdf:type rdfs:Property ;
                rdfs:domain rdfs:Resource ;
                rdfs:range rdfs:Resource .
```

Fig. 3. A fragment of the RDF Mapping Schema in N3.

Figure 4 shows a fragment of the global ontology with the mappings to local schemas (S_1, S_2, and R). The concrete mapping process is ignored for the sake of space. Notice that the mapping `S1/member/name eq (lastname And firstname)` contains two mapping operators: the constraint mapping operator `eq` (i.e., `=`) and the operation mapping operator `And`. In order to enable reformulation of a source query to the target query over a local XML source, we use XML path expressions in the mappings to reflect the nesting structure of the XML source.

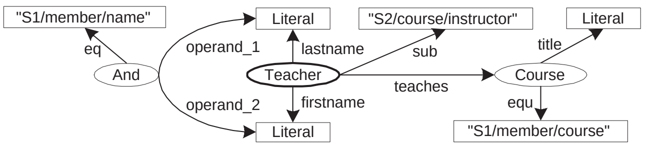

Fig. 4. A fragment of the global ontology with the mappings to local schemas.

4 Conclusions and Future Work

In this paper, we focus on the issue of mapping representation in a framework that integrates XML and RDF sources by using an RDF-based global ontology. We propose a mapping language called RDFMS that is a meta-schema defined on top of RDFS. Rather than giving a complete formal definition of RDFMS, we give an overview of the solution, along with concrete examples. Future work will focus on: (1) Developing a complete mapping language based on this work. (2) Incorporating the mapping representation method into an integration system (e.g., [9]) to improve the query answering process.

References

1. B. Amann, C. Beeri, I. Fundulaki, and M. Scholl. Ontology-Based Integration of XML Web Resources. In *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, pages 117–131, 2002.
2. B. Amann, I. Fundulaki, M. Scholl, C. Beeri, and A. Vercoustre. Mapping XML Fragments to Community Web Ontologies. In *Proceedings of the 4th International Workshop on the Web and Databases (WebDB 2001)*, pages 97–102, 2001.
3. S. Bergamaschi, S. Castano, and M. Vincini. Semantic Integration of Semistructured and Structured Data Sources. *SIGMOD Record*, 28(1):54–59, 1999.
4. T. Berners-Lee. An RDF language for the Semantic Web: Notation 3. http://www.w3.org/DesignIssues/Notation3.html.
5. Y. Bishr. Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4):229–314, 1998.
6. A. Calì, D. Calvanese, G. D. Giacomo, and M. Lenzerini. On the Expressive Power of Data Integration Systems. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002)*, pages 338–350, 2002.
7. S. D. Camillo, C. A. Heuser, and R. S. Mello. Querying Heterogeneous XML Sources through a Conceptual Schema. In *Proceedings of the 22nd International Conference on Conceptual Modeling (ER 2003)*, pages 186–199, 2003.
8. Y. Chen and P. Revesz. CXQuery: A Novel XML Query Language. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science, and Medicine on the Internet (SSGRR 2002w)*, 2002.
9. I. F. Cruz and H. Xiao. Using a Layered Approach for Interoperability on the Semantic Web. In *Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003)*, pages 221–232, 2003.
10. I. F. Cruz, H. Xiao, and F. Hsu. An Ontology-based Framework for Semantic Interoperability between XML Sources. In *Eighth International Database Engineering & Applications Symposium (IDEAS 2004)*, July 2004.
11. A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov. Piazza: Data Management Infrastructure for Semantic Web Applications. In *Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, pages 556–567, 2003.
12. L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernandez, and R. Fagin. Translating web data. In *Proceedings of VLDB*, pages 598–609, 2002.
13. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.