The Role of Ontologies in Data Integration

Isabel F. Cruz Huiyong Xiao

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

Abstract

In this paper, we discuss the use of ontologies for data integration. We consider two different settings depending on the system architecture: central and peer-to-peer data integration. Within those settings, we discuss five different cases studies that illustrate the use of ontologies in metadata representation, in global conceptualization, in high-level querying, in declarative mediation, and in mapping support. Each case study is described in detail and accompanied by examples.

1 Introduction

1.1 Data Integration

Data integration provides the ability to manipulate data transparently across multiple data sources. It 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 [1, 3, 7, 10, 18, 22] and peer-to-peer data integration systems [2, 4, 5, 11, 16, 19]. 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) [22, 17]. 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.

1.2 Data Heterogeneity

Data sources can be heterogeneous in syntax, schema, or semantics, thus making data interoperation a difficult task [6]. Syntactic heterogeneity is caused by the use of different models or languages. Schematic heterogeneity results from structural differences. Semantic heterogeneity is caused by different meanings or interpretations of data in various contexts. To achieve data interoperability, the issues posed by data heterogeneity need to be eliminated.

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). In this paper, we are concerned with solving all three kinds of heterogeneities by bridging syntactic, schematic, and semantic heterogeneities across different sources.

1.3 Semantic Data Integration using Ontologies

We call *semantic data integration* the process of using a conceptual representation of the data and of their relationships to eliminate possible heterogeneities. At the heart of semantic data integration is the concept of *ontology*, which is an explicit specification of a shared conceptualization [13, 14].

Ontologies were developed by the Artificial Intelligence community to facilitate knowledge sharing and reuse [15]. 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. For example, the global schema in a data integration system may be an ontology, which then acts as a mediator for reconciliating the heterogeneities between different sources. As an example of the use of ontologies on peer-to-peer data integration, we can produce for each source schema a local ontology, which is made accessible to other peers so as to support semantic mappings between different local ontologies.

1.4 Paper Overview

We review the use of ontologies on heterogeneous data integration systems. Based on existing approaches to ontology-based data integration and in particular on our work on central and peer-to-peer data integration, we discuss how ontologies can be used to facilitate data interoperation and integration. In Section 2, we present an overview of the concept of ontology and of languages that are used for representing ontologies. In Section 3, we give a high-level description of the use of ontologies in data integration. In the following two sections we discuss five case studies describing typical uses of ontologies. Three of those case studies relate to central data integration and are presented in Section 4. The other two case studies relate to peer-to-peer data integration and are presented in Section 5. We conclude in Section 6.

2 Ontologies

An *ontology* is a formal, explicit specification of a shared conceptualization [13]. 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 an ontology captures consensual knowledge, that is, it is accepted by a group. "Explicit" means that the type of concepts in an ontology and the constraints on these concepts are explicitly defined. Finally, "formal" means that the ontology should be machine understandable.

Typical "real-world" ontologies include taxonomies on the Web (e.g., Yahoo! categories), catalogs for on-line shopping (e.g., Amazon.com's product catalog), and domain-specific standard terminology (e.g., UMLS[1] and Gene

[1] http://www.nlm.nih.gov/research/umls/

Ontology[2]). As an online lexicon database, WordNet[3] is widely used for discovery of semantic relationships between concepts.

Existing ontology languages include:

XML Schema. Strictly speaking, XML Schema is a semantic markup language for Web data. The database-compatible data types supported by XML Schema provide a way to specify a hierarchical model.[4] However, there are no explicit constructs for defining classes and properties in XML Schema, therefore ambiguities may arise when mapping an XML-based data model to a semantic model.

RDF and RDFS. RDF (Resource Description Framework) is a data model developed by the W3C for describing Web resources.[5] RDF allows for the specification of the semantics of data in a standardized, interoperable manner. In RDF, a pair of resources (nodes) connected by a property (edge) forms a statement: (resource, property, value). RDFS (RDF Schema)[6] 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. DAML-OIL (DARPA Agent Markup Language-Ontology Interface Language) is a full-fledged Web-based ontology language developed on top of RDFS.[7] 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 XML Schema data types for semantic interoperability in XML.

OWL. 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.[8]

---

[2] http://www.geneontology.org
[3] http://www.cogsci.princeton.edu/∼wn/
[4] http://www.w3.org/TR/xmlschema-2
[5] http://www.w3.org/TR/rdf-primer
[6] http://www.w3.org/TR/rdf-schema
[7] http://www.w3.org/TR/daml+oil-reference
[8] http://www.w3.org/TR/owl-ref

Other ontology languages include SHOE (Simple HTML Ontology Extensions),[9] XOL (Ontology Exchange Language),[10] and UML (Unified Modeling Language).[11]

Among all these ontology languages, we are most interested in XML Schema and RDFS for their particular roles in data integration and the "Semantic Web" [12]. More specifically, XML Schema and RDFS use the same syntax and can be used for data modeling and ontology representation. But they have their own particular features in the sense that XML data has *document structure* in terms of the nesting elements in an individual XML document, whereas RDF data has *domain structure* formed by the concepts and relationships between concepts [11, 16]. We shall discuss this issue in detail in Section 4.

3 Ontologies for Data Integration

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 [23]:

Single ontology approach. All source schemas are directly related to a shared global ontology that provides a uniform interface to the user [9]. 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. 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, an additional representation formalism is necessary for defining the inter-ontology mappings. The OBSERVER system [18] is an example of this approach.

Hybrid ontology approach. A combination of the two preceding approaches is used. 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

[9]http://www.cs.umd.edu/projects/plus/shoe
[10]http://www.ai.sri.com/pkarp/xol/
[11]http://www.uml.org/

existing mappings. Our layered framework [9] 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. A *hybrid* peer-to-peer system, where a global ontology exists in a "super-peer" can also use the hybrid ontology approach [11]. The multiple ontology approach can be best used to construct *pure* peer-to-peer data integration systems, where there are no super-peers.

We identify the following five uses of ontologies in data integration:

Metadata Representation. Metadata (i.e., source schemas) in each data source can be explicitly represented by a local ontology, using a single language.

Global Conceptualization. The global ontology provides a conceptual view over the schematically-heterogeneous source schemas.

Support for High-level Queries. Given a high-level view of the sources, as provided by a global ontology, the user can formulate a query without specific knowledge of the different data sources. The query is then rewritten into queries over the sources, based on the semantic mappings between the global and local ontologies.

Declarative Mediation. Query processing in a hybrid peer-to-peer system uses the global ontology as a declarative mediator for query rewriting between peers.

Mapping Support. A thesaurus, formalized in terms of an ontology, can be used for the mapping process to facilitate its automation.

In the following sections we discuss five case studies, which correspond to the above five uses. The three first case studies are in the context of centralized data integration systems (Section 4), while the last two are in the context of peer-to-peer data integration systems (Section 5). We base our discussion on our previous work [9, 10, 11, 24, 25].

4 Central Data Integration

In this section, we will describe three case studies of ontologies in the context of central data integration. To make the issues concrete, we use a running example involving two XML sources and demonstrate how to enable semantic interoperation between them.

Example 1 *Figure 1 displays two XML schemas (S_1 and S_2) and their respective documents (D_1 and D_2), which are represented as trees. The two XML documents conform to different schemas but represent data with similar semantics. In particular, both schemas represent a many-to-many relationship between two concepts:* `book` *and* `author` *in S_1 (equivalently denoted by* `article` *and* `writer` *in S_2). However, structurally speaking, they are different: S_1 (book-centric schema) has the* `author` *element nested under the* `book` *element, whereas S_2 (author-centric schema) has the* `article` *element nested under the* `writer` *element.*

Semantically equivalent data elements, such as the authors of publication "b_2", can be reached using different XML path patterns, respectively for schema S_1 and schema S_2:

```
/books/book[@booktitle="b2"]/author/@name
```

and

```
/writers/writer[article/@title="b2"]/@fullname
```

where the contents in the square brackets specify the constraints for the search patterns.

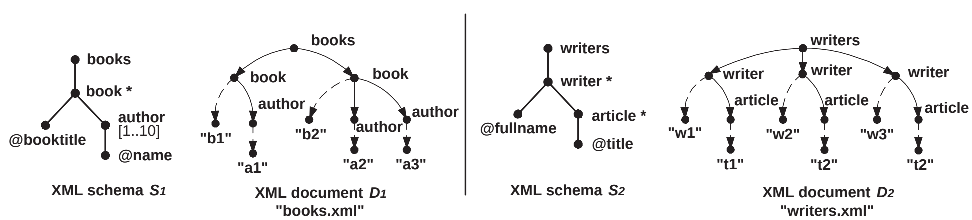

Figure 1: Two XML sources with heterogeneous schemas.

The example demonstrates that multiple XML schemas (or structures) can exist for a single conceptual model. In comparison, the schema or ontology languages (e.g., RDFS, DAML+OIL, and OWL) that operate on the conceptual level are *structurally flat* so that the user can formulate a query from a conceptual perspective without considering the structure of the source [1, 7, 23, 10].

Figure 2 shows the architecture of a system that interoperates among

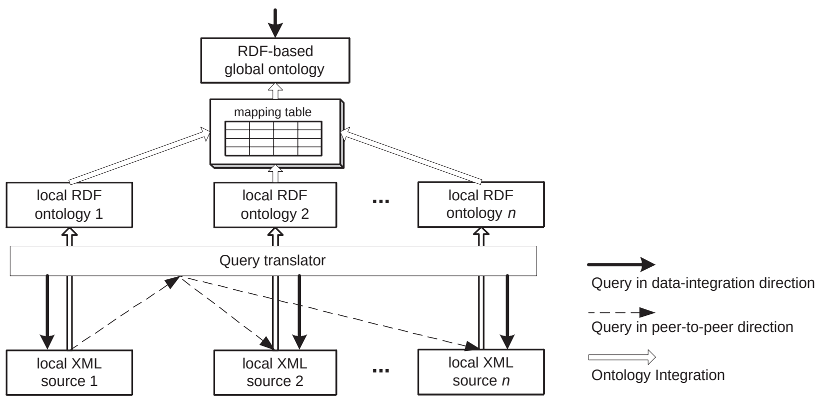

Figure 2: An architecture for XML data integration.

schematically heterogeneous data sources [10]. The following three cases study in detail the principles embodied in this architecture.

Case Study 1 - Metadata Representation

As a first step for bridging across the heterogeneities of diverse local sources, a *local ontology* must be generated from each source database schema (e.g., relational, XML, or RDF). A local ontology is a conceptualization of the elements and relationships between elements in each source schema. To facilitate interoperation, those ontologies should be expressed using the same model. Furthermore, for the sake of correct query processing, the structure of source schemas and the integrity constraints (e.g., relational foreign keys) expressed on the schemas should be preserved in the local ontology. We choose RDFS to represent each local ontology.

In our approach, ontology generation from source schemas is accomplished by *model-based schema transformation* [9]. In particular, the following approaches are taken for the relational and XML schema transformation:

Relational Schema. Relations are converted into RDF classes and attributes into RDF properties, which are attached to the class corresponding to the relation to which the attributes belong. Foreign key dependencies between two relations are represented by two properties (corresponding to the two relations) sharing the same value in the target local ontology.

XML Schema. Complex-type elements are converted into RDF classes and simple-type elements and attributes are converted into RDF properties. This transformation process encodes the mapping information between each concept in the local RDF ontology and the path to the corresponding element in the XML source. Nesting relationships between XML elements are represented using a meta-property rdfx:contains; rdfx stands for the namespace where contains is defined. This meta-property enables the RDF representation of the XML nesting structure, by connecting two RDF classes representing the two nesting XML elements.

Example 2 *Following Example 1, Figure 3 shows the local RDF ontologies S_1' and S_2', which are generated respectively from the XML source schemas S_1 and S_2.*

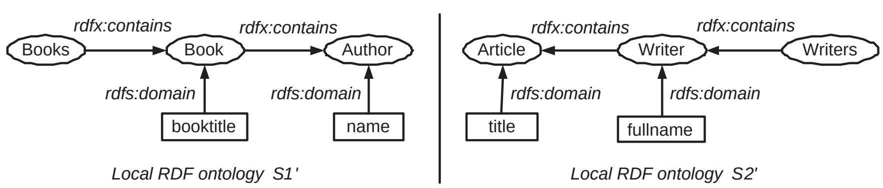

Figure 3: RDF-based local ontologies generated from XML schemas.

Case Study 2 - Global Conceptualization

To make the integration system accessible through the uniform interface of the global ontology, semantic mappings are established between the global ontology and the local ontologies. In our approach, this mapping process is accomplished during the construction of the global ontology, which is generated by merging the local ontologies, for example, using a GaV approach.

We consider that each local ontology is merged into the global ontology, the *target ontology*. The process of ontology merging consists of several operations:

- *Copying a class and/or its properties:* classes and properties that do not exist in the target ontology are copied into it.
- *Class Merging:* conceptually equivalent classes in the local and target ontologies are combined into one class in the target ontology.
- *Property Merging:* conceptually equivalent properties of a class in the local and target ontologies are combined into one property in the target ontology.
- *Relationship Merging:* conceptually equivalent relationships from one class c_1 to another class c_2 in the local and target ontologies are combined into a single relationship in the target ontology (i.e., an RDF property having c_1 as its domain and c_2 as its range).
- *Class Generalization:* related classes in the local and target ontologies can be generalized into a a superclass. The superclass can be obtained by searching an existing knowledge domain (e.g., the DAML Ontology Library [12]) or reasoning over a thesaurus.

We note that along with the above operations, semantic correspondences are established. For example, for each element p_L in a local ontology, if there exists a semantically equivalent element p_G in the global ontology, the two elements will be merged and a correspondence between p_L and p_G will be generated.

Example 3 *Figure 4 shows the global RDF ontologies generated by merging the local ontologies S'_1 and S'_2 of Example 2. Note that the classes (properties) represented in grey are merged classes (properties), and the classes* Book *and* Author *are also extended, with* Publication *and* Person *being their superclasses, respectively.*

Case Study 3 - Support for High-level Queries

Given a conceptual view of available information sources, the user may pose a query in terms of the global ontology. We say the query is a *high-level* query if its formulation does not require awareness of particular source schemas. The query is then reformulated by a rewriting algorithm into a

[12]http://www.daml.org/ontologies/

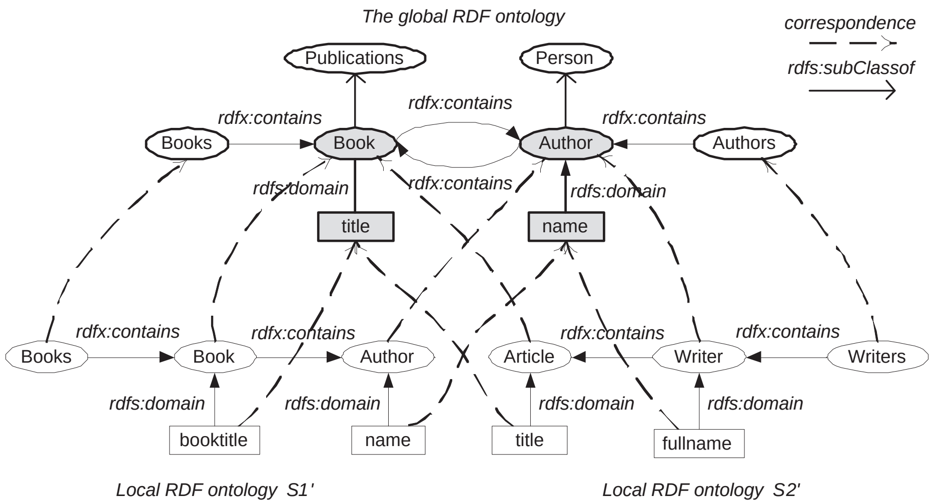

Figure 4: A conceptual view on local sources.

subquery for each source. The subqueries over sources are subject to the structure of source schemas, and may be expressed in a different language from that of the high-level query. An inference mechanism may be needed in the query rewriting, for example, when a concept involved in the query has super-concepts or sub-concepts.

In addition to handling high-level queries on the global ontology, a *bidirectional* query translation algorithm is also supported [10] (see Figure 2). In this case, we can translate a query posed against an XML source to an equivalent query against any other XML source.

Example 4 *Suppose the user asks the query "Find the persons who have written publication b_2." This query will be expressed in a RDF query language such as RDQL.* [13] *First,* Person *has sub-concept* Author*, which corresponds to two different concepts (*Author *and* Writer*) in two different RDF local databases. Therefore the initial query will be rewritten as two sub-queries to those databases. In turn, those queries may be further rewritten using a XML query language incorporating the path expressions of Example 1 (unless the data was materialized under the RDF local ontologies). Using the bidirectional query translation mechanism, a query involving the concepts* Book *and* Author *in one source will be translated into a query involving* Article *and*

[13] http://www.hpl.hp.com/semweb/rdql.htm

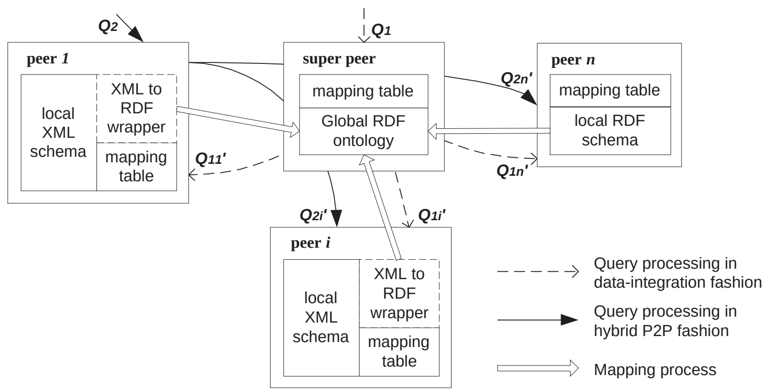

Figure 5: The hybrid peer-to-peer architecture of PEPSINT.

Writer *in another data source, by using the correspondences established by the global ontology.*

5 Peer-to-Peer Data Integration

We consider again the two XML sources of Figure 1. However, this time they are connected in a peer-to-peer architecture. We consider a *hybrid* peer-to-peer architecture with two types of peers: *super-peers* containing the global RDF ontology, and *peers* each containing a data source and an ontology. Each peer represents an autonomous information system and connects to a super-peer via semantic mappings. Peer-to-peer data integration systems or frameworks include LRM (Local Relational Model) [5], Hyperion [2], Piazza [16], PeerDB [19], SEWASIE [4], and PEPSINT [11].

Case Study 4 - Declarative Mediation

The PEPSINT system is a hybrid peer-to-peer system whose architecture is shown in Figure 5. PEPSINT uses a GaV approach. The global ontology in a super-peer serves two functions: (1) It provides the user with a uniform high-level view of the data sources in the distributed peers, and (2) it serves as a a mediator for query translation from one peer to another. The former function is similar to the one described in Case Study 3. The latter function is discussed in detail here.

The user can pose a query against the local XML or RDF data source in any peer. Locally, the query will be executed on the local source to get a *local*

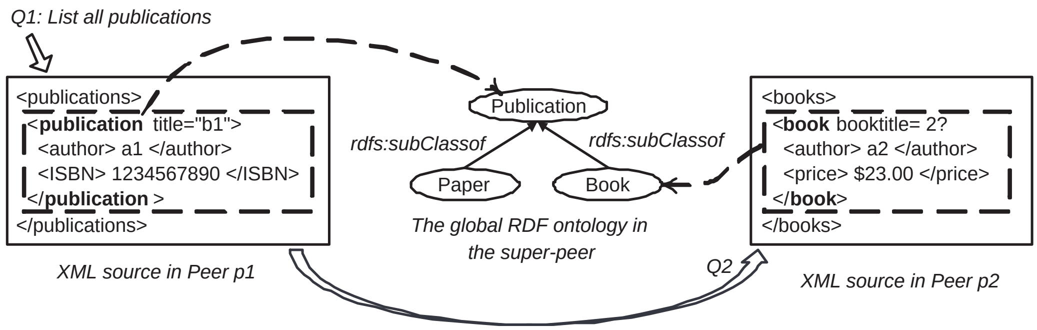

Figure 6: Mediation for peer-to-peer query rewriting.

answer. Meanwhile, the source query is rewritten into a target query over every connected peer. The query rewriting utilizes the global ontology, and the composition of mappings from the original peer to the super peer with mappings from the super-peer to the target peers. By executing the target query, each peer returns an answer to the original peer, called the *remote answer*. The local and remote answers are integrated and returned to the user at the site of the originating peer.

Example 5 *Consider two XML sources, one in peer p_1 and the other in peer p_2, and a global ontology expressed in RDF in a super-peer. As shown in Figure 6, the global ontology consists of a class* Publication *and two subclasses* Paper *and* Book*. The* Publication *class is mapped to the* publication *element of the XML source in p_1, while the class* Book *corresponds to* book *of the XML source in p_2. An XML query Q_1 on p_1 involving* publication *will be rewritten to a target query Q_2 on p_2 involving include* book*. The XML fragments inside the dashed-line boxes are integrated and returned as answers.*

Case Study 5 - Mapping Support

A thesaurus can be used for data integration to facilitate the automation of the schema mapping process [21, 9]. In particular, it can help discovering the semantic relationships between concepts in different schemas or ontologies. WordNet is an example of such a thesaurus. It consists of a network of terms and their *semantic relations* (e.g., synonym, hypernym, and hyponym). A term may have multiple senses, each being a *synset*.

A thesaurus-based schema matching approach has been devised for peer-to-peer data integration [24]; this approach consists of the following three steps (as illustrated in Figure 7):

1. Path Exploration. Among the semantic relations between synsets in WordNet, we choose those of *synonymy*, *hyponymy/hypernymy* (i.e., more specific/more general), and *related-to*, when enumerating the paths between two arbitrary concepts from different local ontologies in peers. As shown in Figure 7, six paths are found from Quantity to Number.

2. Path Selection. When multiple paths are found between two concepts, we choose the *optimal path*, which corresponds to the most likely semantic relation between the two concepts. For this purpose, *semantic similarities* (i.e., the number above each path in the figure) are calculated for all the paths. The calculation is implemented by assigning different semantic relations with different *weights* (e.g., 1 for synonymy and 0.8 for hypernymy) and then taking the average of all the weights. The path with highest similarity is then chosen as the optimal path. If there is more than one such path, then the user's intervention is needed.

3. Semantic Derivation. The last step is to derive the (direct) semantic relationship, *Sem*, between the two concepts by reasoning on the semantic relations along the optimal path p between them. More specifically, $Sem(p) = Sem(p_n)$ is computed based on the following recursive algorithm, where $p_n = (r_1, r_2, ..., r_n)$, and $r_i (1 \leq i \leq n)$ are the edges (semantic relations) along p.

$$Sem(p_n) \quad = Sem(p_{n-1}) \wedge Sem(r_n), \qquad if\ n > 1; \tag{1}$$

$$Sem(p_n) \quad = \quad \approx,\ \supseteq,\ \subseteq,\ or\ \sim, \qquad if\ n = 1. \tag{2}$$

In the above formulas, the symbols $\approx$, $\supseteq$, $\subseteq$, and $\sim$, respectively stand

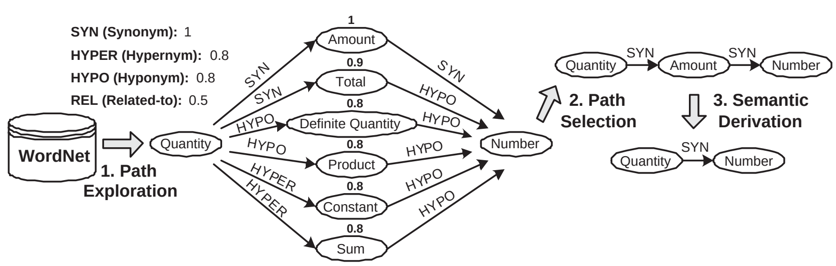

Figure 7: Thesaurus-based schema mapping process.

for the semantic relation of synonymy, hypernymy, hyponymy, and related-to. The operation $\wedge$ obeys the rules that are shown in Table 1.

$\wedge$	$\approx$	$\supseteq$	$\subseteq$	$\sim$
$\approx$	$\approx$	$\supseteq$	$\subseteq$	$\sim$
$\supseteq$	$\supseteq$	$\supseteq$	?	$\sim$
$\subseteq$	$\subseteq$	?	$\subseteq$	$\sim$
$\sim$	$\sim$	$\sim$	$\sim$	$\sim$

Table 1: Inference rules for semantic relations: a white cell (at the intersection of each pair of grey cells) contains the result of the operation on the relations in the two grey cells, and a question mark indicates that human intervention is needed.

6 Conclusions

The advent of XML has created a syntactic platform for Web data standardization and exchange. However, XML has several problems. First of all, documents expressed in XML share the same syntax, but can be otherwise heterogeneous, for example by having different structures and naming conventions. Also, an XML document does not express the semantics of the elements or of the relationships among elements explicitly. Therefore, it is not a suitable language for metadata representation.

Ontologies provide an explicit and formal specification of a shared conceptualization, and are able to facilitate knowledge sharing and reuse. We use ontologies expressed in RDFS, a semantically rich schema language, to bridge across syntactic, schematic, and semantic heterogeneities in data sources.

In this paper, we have presented five different case studies that illustrate the role that ontologies play in the process of data integration, in centralized and peer-to-peer architectures.

Related research includes research on ontology generation, ontology mapping, and ontology evolution. An ontology can be generated manually using an authoring tool or (semi-)automatically from various knowledge sources (e.g., database schemas). Techniques used for ontology mapping, including ontology alignment and ontology merging [20, 8], overlap to a large extent with those techniques for schema matching [21]. Finally, ontology evolution, also called ontology versioning, involves changes on representation, structure, and semantics of ontologies. Each step of such an evolution must ensure the consistency between the old version and the improved version of the ontology, just as if a database schema's evolution must guarantee the consistency of the new schema with the data.

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] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. J. Miller, and J. Mylopoulos. The Hyperion Project: From Data Integration to Data Coordination. *SIGMOD Record*, 32(3):53–38, 2003.

[3] Y. Arens, C. A. Knoblock, and C. Hsu. Query Processing in the SIMS Information Mediator. In *The AAAI Press*, May 1996.

[4] S. Bergamaschi, F. Guerra, and M. Vincini. A Peer-to-Peer Information System for the Semantic Web. In *Proceedings of the International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2003)*, July 2003.

[5] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data Management for Peer-to-Peer Computing: A Vision. In *WebDB 2002*, pages 89–94, 2002.

[6] Y. A. Bishr. Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4):229–314, 1998.

[7] S. D. Camillo, C. A. Heuser, and R. dos Santos 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] I. F. Cruz, W. Sunna, and A. Chaudhry. Semi-Automatic Ontology Alignment for Geospatial Data Integration. In *GIScience 2004*, LNCS, pages 51–66. Springer Verlag, 2004.

[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, Rome, Italy, December 2003.

[10] I. F. Cruz, H. Xiao, and F. Hsu. An Ontology-based Framework for Semantic Interoperability between XML Sources. In *Proceedings of the 8th International Database Engineering & Applications Symposium (IDEAS 2004)*, pages 217–226, July 2004.

[11] I. F. Cruz, H. Xiao, and F. Hsu. Peer-to-Peer Semantic Integration of XML and RDF Data Sources. In *The 3rd International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2004)*, July 2004.

[12] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. C. A. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing*, 4(5):63–74, 2000.

[13] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[14] T. R. Gruber and G. R. Olsen. An Ontology for Engineering Mathematics. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR 1994)*, pages 258–269, 1994.

[15] N. Guarino. Formal Ontology and Information Systems. In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS 1998)*, pages 3–15, 1998.

[16] 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.

[17] M. Lenzerini. Data Integration: A Theoretical Perspective. In *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002)*, pages 233–246, Madison, Wisconsin, June 2002. ACM.

[18] E. Mena, V. Kashyap, A. P. Sheth, and A. Illarramendi. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. In *Proceedings of the 1st IFCIS International Conference on Cooperative Information Systems (CoopIS 1996)*, pages 14–25, 1996.

[19] W. S. Ng, B. C. Ooi, K. L. Tan, and A. Zhou. PeerDB: A P2P-based System for Distributed Data Sharing. In *Proceedings of the 19th International Conference on Data Engineering (ICDE 2003)*, pages 633–644, 2003.

[20] N. F. Noy and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 2000)*, pages 450–455, 2000.

[21] E. Rahm and P. A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB J.*, 10(4):334–350, 2001.

[22] J. D. Ullman. Information Integration Using Logical Views. In *Proceedings of the 6th International Conference on Database Theory (ICDT 1997)*, pages 19–40, 1997.

[23] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-Based Integration of Information - A Survey of Existing Approaches. In *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, 2001.

[24] H. Xiao and I. F. Cruz. RDF-based Metadata Management in Peer-to-Peer Systems. In *The 2nd IST Workshop on Metadata Management in Grid and P2P System (MMGPS 2004)*, 2004.

[25] H. Xiao, I. F. Cruz, and F. Hsu. Semantic Mappings for the Integration of XML and RDF Sources. In *Workshop on Information Integration on the Web (IIWeb 2004)*, August 2004.