RDF-based Metadata Management in Peer-to-Peer Systems

Huiyong Xiao   Isabel F. Cruz

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

Abstract. Peer-to-Peer (P2P) data integration combines the P2P infrastructure with traditional schema-based data integration techniques. One of the primary problems in this research area is the automation of the process of interoperation among heterogeneous data sources. To facilitate this process, metadata are used to add rich semantics to the data sources. In this paper, we propose an RDF-based layered approach to data interoperation in a P2P environment. We use RDF and its vocabulary language RDFS for representation of data and metadata, which are distributed in different peers. For the representation of the mappings of metadata from peer to peer (called *P2P mappings*), we propose a mapping language, which is an RDF-based meta-ontology (called *RDFMS*) that enables metadata management in a decentralized way. We also discuss a thesaurus-based mapping process and query translation in the proposed framework.

1 Introduction

Recently, Peer-to-Peer (P2P) techniques are becoming more and more popular in various applications, for its properties such as decentralization, autonomy, and scalability [15]. Typical examples include Napster (www.napster.com) and Gnutella (www.gnutella.com) for file sharing, SETI@home (setiathome.ssl.berkeley.edu) and Grid (www.grid.org) for distributed computing, and AOL instance messaging (www.aol.com) for collaborative work. As an emerging application area, P2P data integration combines the P2P infrastructure with traditional schema-based data integration techniques [1, 3, 6, 7, 11, 16, 17]. For this purpose, one has to first solve the problem of interoperation among heterogeneous data sources. To facilitate automation of the process of data interoperation, metadata (typically in a uniform representation) are used to add rich semantics into data sources. Furthermore, a powerful mechanism is needed for expressing and managing the metadata to support complex queries over the resources in the network.

The example in Figure 1 illustrates the problem more intuitively. The three schema-based data sources (residing in different peers) are heterogeneous in both syntax and structure. The data in Peer *p1* contains an XML schema in DTD

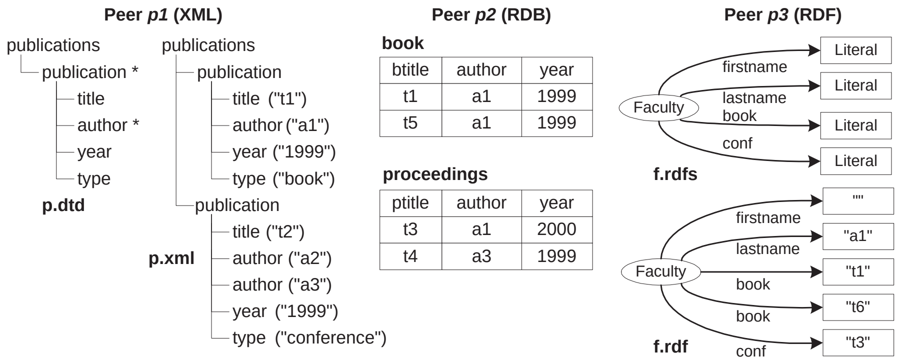

Fig. 1. A motivating example of P2P data interoperation.

(`p.dtd`) and its data (`p.xml`), where the strings in parentheses are element or attribute values. Peer *p2* is a relational database (RDB) with two relations (`book` and `proceedings`). Peer *p3* contains RDF data (`f.rdf`) with its RDF schema (`f.rdfs`) defined in RDF Schema[1] (RDFS, a vocabulary language for RDF). One notable structural difference among these data sources is that the semantically equivalent terms are formulated in different forms. That is, the two types of publications `book` and `proceedings` are designed as values (instances) in *p1*, as relation names in *p2*, and as property names in *p3*.

The metadata for each data source in the example is a schema, which defines the structure of the data. To enable data access across these data sources, semantic mappings have to be established to hide the syntactic and structural differences. As an example, our previous work has discussed this problem and provided for it a *hybrid* P2P architecture, in which a *super peer* is used as a central control point [10]. In this paper, we propose an RDF-based layered approach to data interoperation in a *pure* P2P environment, in which no central points exist and all peers play a similar role.

We base our approach on RDF (through the respective use of RDFS and a meta-ontology RDFMS for representation of local metadata and inter-schema mappings) for its particular characteristics as a mechanism for Web resource description. First of all, RDFS organizes metadata declaratively and conceptually (i.e., independently of the document structure) in terms of classes, properties, and data types. Using the metadata, the semantics of concepts and semantic relationships between concepts can be explicitly expressed, thus providing a good foundation for reasoning on the subsequent metadata mapping and query reformulation. Second, RDF properties are actually binary relations between concepts. This feature supports well the representation of P2P mappings, which in our approach are defined as correspondences (other than queries) between metadata in different peers. Finally, a feature of RDF is its extensibility, which allows

[1] http://www.w3.org/TR/rdf-schema/

for easily adding new constructs (e.g., a new mapping type) into our proposed RDFMS-based mapping language.

Our contributions in this paper include: (1) an RDF-based meta-ontology, *RDFMS* (RDF Mapping Schema), which provides a generic way to representing *P2P mappings* between RDF schemas, thus enabling metadata management in a decentralized way, (2) a thesaurus-based schema matching for facilitating automatically computing such P2P mappings, and (3) a preliminary proposal for P2P query answering based on a uniform first-order relation based interpretation. The advantages of these proposed methods are discussed in detail in subsequent sections, respectively.

The rest of the paper is organized as follows. In Section 2 we describe exiting related work. The layered architecture is introduced in Section 3. In Section 4 we discuss metadata representation in RDF, to which XML and relational schemas are converted. We also discuss in detail the P2P mapping process and mapping representation using RDFMS. The process of query translation based on the P2P mappings is given in Section 5. Finally, Section 6 concludes and discusses future work.

2 Related Work

There are already a number of P2P data management systems or frameworks addressing metadata management in the P2P context.

AXML (ActiveXML) [1] proposes a framework that harnesses *web services* for P2P data integration. In AXML, the mapping process and query processing among peers are respectively realized by *embedding* calls to web services in an XML document and by *firing* these web service calls. In comparison with the schema-based approach proposed in this paper, AXML is approaching the goal of P2P integration at the data level.

LRM (Local Relational Model) [7], **Hyperion** [3], and **Piazza** [11] provide solutions to P2P metadata and data management, by using *domain relations*, *mapping tables* or *expressions*, and an XQuery-based mapping language, respectively. Compared to these approaches, we believe the RDFMS-based mapping language proposed in this paper has the advantages of maintainability (via its meta-ontology for mappings representation), extensibility (via the reification property of RDF), inference (via declarative and explicit semantics of RDF), and support for query processing (through RQL, which can be used to query the mappings).

Edutella [16] and **PeerDB** [17] implement metadata and data management in a dynamic (i.e., run-time) way. That is, there are no established mappings between peers at design time. Instead, Edutella uses a Datalog-based query exchange language serving as a common query interchange format, and a wrapper is used to map local query languages (e.g., SQL or XPath) into the common format. In PeerDB, query reformulation between peers is assisted by agents through a *relation-matching strategy*, a process of matching the metadata of RDBs in different peers.

In **SEWASIE** [6], each peer contains an information node (SINode) that is supported by MOMIS [5], which is a mediator-based system. Within each SINode, semantic mappings between the local schemas and a *global virtual view* are established in the Global-as-View (GAV) approach. A preliminary discussion on schema mapping representation (also in the GAV approach) using RDF is provided [20], from which the RDFMS language with the meta-ontology that we propose in this paper are developed.

3 The Layered Peer Architecture

In a P2P data management system, a peer is supposed to be able to manage its local data source as in a traditional database system. In addition, a peer has to possess the ability of communicating with the other peers by providing and consuming services. For this purpose, we propose for each peer a layered architecture (as shown in Figure 2), by which distributed peers form a pure P2P network.

The peer architecture consists of four layers, in which each upper layer achieves its functionality based on the lower ones. In particular, the *syntax layer* provides a uniform syntax (RDF/XML) for serializing the local ontology and its data, which are converted from the local source residing the peer by a wrapper. The *representation layer* is responsible for representing the local ontology in RDFS and its mappings in RDFMS. The *service layer* implements both schema mapping and the processing of queries, which are two of the main services that a peer provides. The *application layer* contains a GUI (Graphic User Interface) for interaction with the user, which initiates query requests. In summary, the adoption of the layered peer architecture simplifies the resolution of peer-to-peer heterogeneities into level-to-level dependencies, thus facilitating the data interoperation process by facilitating the maintenance of the layers and their reusability. We describe next the key components of the layers.

XML/RDB Wrapper. A wrapper converts the data in the peers into a uniform representation in RDF before they are made accessible to other peers. In our approach, the XML and RDB wrappers are respectively used to transform

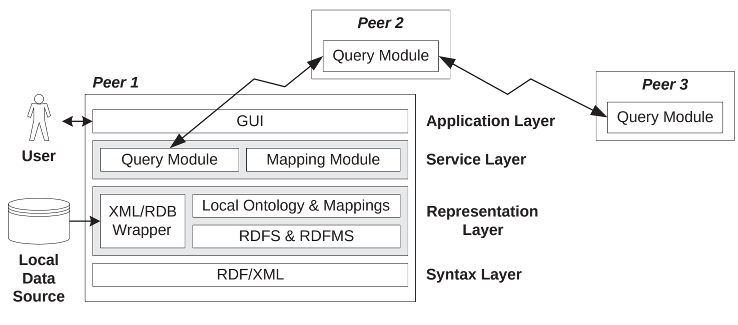

Fig. 2. The layered peer architecture.

XML schemas and relational schemas into local ontologies in RDF. To facilitate local query execution, the local data is also converted and serialized (in the syntax layer) and may be managed using a local RDF database (e.g., RSSDB [2]).

Mapping Module. The mapping module is responsible for establishing P2P semantic mappings based on a thesaurus (WordNet[1] in our approach), which is a lexicon database providing semantic clues for concepts and their relationships. More specifically, the mapping module takes as input the two local ontologies (already represented in RDF), finds in the thesaurus a semantically optimal path between each pair of elements in different local ontologies, and derives the direct semantic relationship between elements based on certain rules. This P2P schema matching process and the representation of mappings in RDFMS are discussed in greater detail in Section 4.

Query Module. The query module is responsible for query processing, including query execution, query translation, and answer integration. Through the GUI, the user poses a query, called *local query*, using RQL [12]. The query module then interprets the query and executes it over the *host peer*. Meanwhile, the query is also forwarded to each of the linked peers (called *guest peer*), where the query is rewritten into a *remote query* that is executed locally and propagated further. The query translation utilizes the P2P mapping information established by the mapping module and stored in the representation layer. Finally, the *remote answer* (from the guest peers) and the *local answer* (from the host peer) are returned to the host peer and integrated there as a final answer.

4 RDF-based Metadata Management

4.1 Metadata Representation in RDF

As mentioned previously, to achieve a uniform metadata and data representation in the P2P network, wrappers are used to transform heterogeneous sources into RDF. A few methods (e.g., [8, 13]) have been proposed for translating data back and forth between different paradigms (e.g., relational, XML, and RDF). However, they either do the translation at the instance level (instead of at the the schema level), or lack generic translation rules. We use in our framework a generic but simple algorithm for schema and data translation, which is proposed in our previous work [10].

In summary, we convert relations into classes and attributes into properties for translation from an RDB to an RDF-based ontology. For translation from XML to RDF, we convert complex-type elements into classes and simple-type elements (with no subelements, only character contents) and attributes into properties. Figure 3 shows the results of schema translation of the three data sources in the example of Figure 1. For the subsequent thesaurus-based schema matching, names in the local ontologies are *standardized*, e.g., from `ptitle` to `title`.

[1] http://www.cogsci.princeton.edu/∼wn/

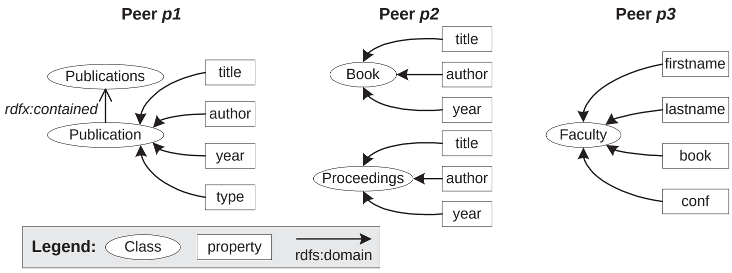

Fig. 3. The RDF-based local ontologies in peers.

In the instance (data) translation that is based on the schema (metadata) translation, important constraints of the local source should be preserved. For example, *foreign key dependencies* between two relations can be translated as a property shared by two classes (corresponding to the two relations) in the target RDF instance. *Nesting constraints* between XML elements can also be preserved by using a new user-defined metaproperty `rdfx:contained` (`rdfx` stands for the namespace), which connects two RDF classes representing the two XML elements [10].

4.2 Thesaurus-based P2P Schema Matching

Schema matching is a long-standing issue in the database community since the 80's. Rahm *et al.* have presented a comprehensive survey of existing schema matching approaches [19]. Inspired by the discussion about discovery of lexical relationships in WordNet [4], we propose in this paper a thesaurus-based schema matching approach for P2P data integration, which utilizes WordNet for discovery of semantic relationships between any two concepts. WordNet is a lexicon database, in which terms are organized into a graph with each node being a sense (a term may have multiple senses, each being a *synset*, i.e., a set of synonyms) and edges being semantic relations between senses. The WordNet-based schema matching consists of the following three steps (shown in Figure 4).

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 but standardized concepts from different local ontologies in peers. In our running example, 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 is supposed to reflect 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, etc.) and then taking the average of all the weights. The path with highest similarity is then chosen as the

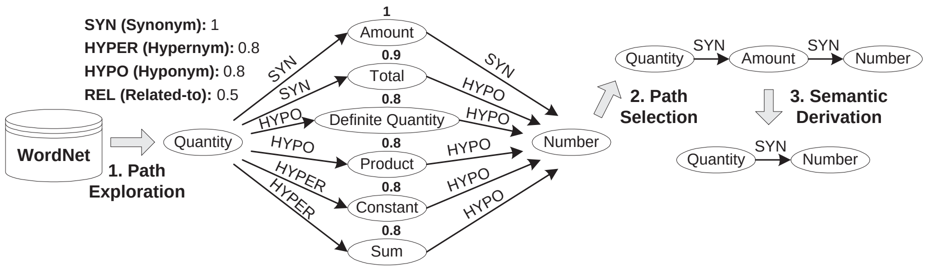

Fig. 4. An example of schema matching using WordNet.

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.

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

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

In the above formulas, the symbols $\approx$, $\supseteq$, $\subseteq$, and $\sim$ respectively stand for semantic relation of synonymy, hypernymy, hyponymy, and related-to. The operation $\wedge$ obeys the rules that are shown in Figure 5, in which a white cell (at the intersection of each pair of gray cells) contains the result of operation on the relations in the two gray cells. Note that a question mark means that human interference is needed.

The above three steps on each pair of elements in two local ontologies that are being mapped determines the inter-schema mappings. In comparison to other thesaurus-based schema matching methods (e.g., [5]), our approach is more general, by providing a way of computing semantic relations between two arbitrary concepts, which may not be directly semantically related.

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

Fig. 5. Inference rules on semantic relations.

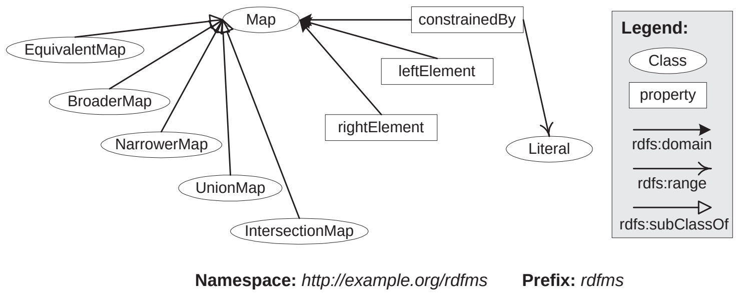

Fig. 6. The meta-ontology of RDFMS.

4.3 RDFMS – A Mapping Language

In this subsection, we discuss the issues related to mapping *types* (i.e., semantics), mapping *cardinality* (e.g., one-to-one or one-to-many mappings), and mapping representation. Every P2P ontology mapping is an instance of the RDFMS meta-ontology.

RDFMS Meta-ontology. To represent the different semantic relationships between concepts in different local ontologies, RDFMS provides such types of mappings as `EquivalentMap` (for synonymy and related-to relations), `BroaderMap` (for hypernymy relations), and `NarrowerMap` (for hyponymy relations). Regarding the case of one-to-many mappings, RDFMS defines `UnionMap` and `IntersectionMap` respectively for two types of logic combinations of the elements on the multiple-element side. Figure 6 shows the meta-ontology represented in the RDF graph model.

In the RDFMS meta-ontology, all types of mappings are defined as classes inheriting from a common class `Map`, which has three general properties that are also inherited by its subclasses. The `leftElement` and `rightElement` properties are used to connect the mapped elements. In order to represent the mapping expression [19] that a P2P mapping may carry, the property `constrainedBy` is defined, whose data type is specified as `Literal`. Following the example in Figure 3, in Figure 7 we show the P2P mappings among the three local ontologies in different peers as represented in RDFMS.

Mapping Atoms. In order to use the mapping information for reasoning on mapping and querying, we define a set of mapping atoms as follows.

- EM(c_1, c_2) iff there exists an instance m of `EquivalentMap`, such that $c_1 = m$.`leftElement` and $c_2 = m$.`rightElement`.
- BM(c_1, c_2) iff there exists an instance m of `BroaderMap`, such that $c_1 = m$.`leftElement` and $c_2 = m$.`rightElement`.
- NM(c_1, c_2) iff there exists an instance m of `NarrowerMap`, such that $c_1 = m$.`leftElement` and $c_2 = m$.`rightElement`.
- UM(c_1, c_2) iff there exists an instance m of `UnionMap`, such that $c_1 = m$.`leftElement` and $c_2 = \{x | x = m$.`rightElement`$\}$, or $c_1 = \{x | x = m$.`rightElement`$\}$ and $c_2 = m$.`leftElement`.

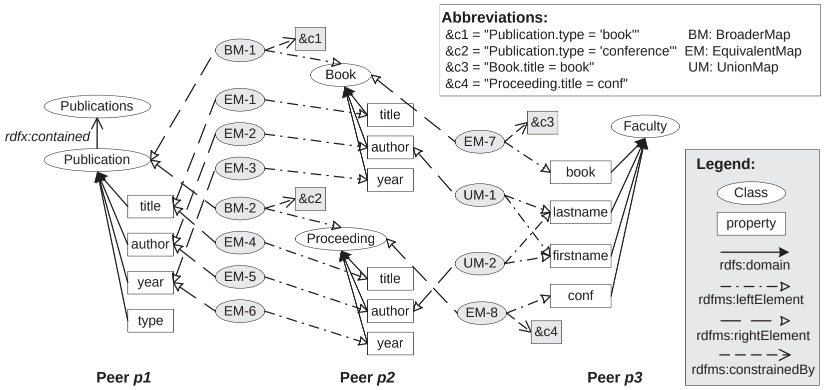

Fig. 7. An example of P2P mappings in RDFMS.

- $\mathsf{IM}(c_1,\ c_2)$ iff there exists an instance m of `IntersectionMap`, such that $c_1 = m.$`leftElement` and $c_2 = \{x|x = m.$`rightElement`$\}$, or $c_1 = \{x|x = m.$`rightElement`$\}$ and $c_2 = m.$`leftElement`.
- $\mathsf{CON}(m,\ e)$ iff given an instance m of `Map` or its subclasses, $e = \{x|x = m.$`constrainedBy`$\}$.

It is easy to derive new mapping types (e.g., *sibling map* or *disjoint map*) and more complex mappings (e.g., many-to-many mappings by combining two `UnionMaps`), based on these atoms. For example, to add a new mapping type of *sibling*, SM, we could simply define such an atom as $\mathsf{SM}(c_1,\ c_2) \equiv \mathsf{NM}(c_1,\ c_3) \wedge \mathsf{NM}(c_2,\ c_3)$. In addition to the above syntax for RDFMS, we also need to define its semantics, which is implicitly used in the query example of Section 5.

Other Mapping Languages. RDFT [18] is proposed also as a meta-ontology for mapping representation, which is ad-hoc w.r.t the case of business integration. Lehti *et al.* propose an OWL-based model particularly for XML data integration [14]. It uses for mapping representation OWL constructs[1] such as `sameClassAs`, `samePropertyAs`, `subClassOf`, and `subPropertyOf`, thus having a limited power, in that they cannot express, for example, cardinality.

5 Queries across the P2P Network

In the syntax layer of our layered peer architecture, we may use an RDF repository (e.g., RSSDB [2]) to store the data converted from local sources. The user then may pose RDF queries over the repository using native languages (e.g., RQL). The basic procedure of query processing (implemented in the query module of a peer) across the P2P network has been described in Section 3. In this section, we focus on the implementation of query translation from peer to peer.

[1] http://www.w3.org/2001/sw/WebOnt/

RQL. RQL is a typed functional RDF query language based on a formal model for directed labeled graphs, which permits the interpretation of superimposed resource descriptions by means of one or more RDF schemas [12]. The basic form of RQL queries for retrieving RDF data is of a SFW expression consisting of `SELECT`, `FROM`, and `WHERE` clauses. For example, the local RQL query Q on Peer p_1 for "listing the titles of all the publications of author a1" may be formulated as follows.

```
SELECT  Y
FROM    {X}title{Y}, {X}author{Z}
WHERE   Z="a1"
```

Most RQL queries are in fact conjunctive queries (called RQL conjunctive queries), which are in the form of $ans(\bar{X})\ :-\ C_1, ..., C_n$, where each C_i, $1 \leq i \leq n$ is an RQL class or a property *pattern* and $\bar{X}$ is a tuple of variables or constants [9].

P2P Query Translation. A first-order relation based model, called SWIM, has also been proposed [9], which we use for the interpretation of RQL conjunctive queries and mapping atoms that are described in Section 4. Because of space limitations, the detailed rules for the interpretation are not discussed here.

Starting from *host peer* p_1, the *local query* (i.e., Q in the example) is propagated across the P2P network. The query module in each peer p_i ($i \neq 1$) is responsible for translating an incoming query into a local query, executing it on the local RDF repository, and forwarding it to all the other linked peers. The query translation can be seen as a function $Q' = f(Q, M)$, where Q, M, and Q' are respectively the incoming query, the mapping information between p_1 and p_i, and the resulting local query. In our approach, the computation of f is simply obtained by composing the underlying SWIM first-order relations of Q with those of M.

Continuing with the running example of this paper, below we show the result of query translation from Q to Q' (in Peer p_2) and to Q'' (in Peer p_3). Note that for the two local schemas in Peer p_2 (see Figure 3), we get the same query Q', which will be executed over the RDF repository of p_2, whereas Q'' contains two different queries, which are executed over the RDF data repository in Peer p_3. By integrating all the answers returned from all peers, the final answer to Q is the set {`"t1", "t3", "t5", "t6"`}.

Q' :
```
SELECT  Y
FROM    {X}title{Y}, {X}author{Z}
WHERE   Z="a1"
```

Q'' :
```
SELECT  Y
FROM    {X}book{Y}, {X}firstname{Z1}, {X}lastname{Z2}
WHERE   Z1="a1" OR Z2="a1"

SELECT  Y
FROM    {X}conf{Y}, {X}firstname{Z1}, {X}lastname{Z2}
```

```
WHERE   Z1="a1" OR Z2="a1"
```

6 Conclusions and Future Work

In this paper, an RDF-based layered approach is proposed for data interoperation in a heterogeneous P2P environment. RDF techniques are used throughout the proposed framework, including RDFS for metadata representation, RDFMS (a mapping language and RDF-based meta-ontology) for the representation of P2P mappings, and RQL for query formulation.

At the center of this paper is a thesaurus-based mapping approach that is used to manage metadata in P2P networks. We discuss the process of schema matching, the RDFMS meta-ontology, and the process of reasoning on the mappings. We also describe briefly the query translation process. We note that we could have used OWL to represent the semantic mappings [14], however, the mixture of constructs for the data and the mappings appears to be less suitable for reasons of maintenance, extensibility, and inference.

The approach proposed in this paper is still a first step to a comprehensive solution to P2P data management. A number of research issues remain, including: (1) The application of built-in constraints of the RDFS model and those defined in the RDFMS model for query processing. (2) The use of inferencing as applied to mapping and to query processing. (3) The study of the complete syntax and semantics of the mapping language. All these research issues together with the implementation of the framework are our targets in future work.

7 Acknowledgments

We would like to thank Eduard Dragut for his suggestions. This research was supported in part by the National Science Foundation under Awards ITR IIS-0326284 and EIA-0091489.

References

1. S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo, and R. Weber. Active XML: Peer-to-Peer Data and Web Services Integration. In *VLDB 2002 (demo)*, pages 1087–1090, 2002.
2. S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, and K. Tolle. The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases. In *Proceedings of the 2nd International Workshop on the Semantic Web (SemWeb 2001)*, 2001.
3. 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.
4. D. Beeferman. Lexical Discovery with an Enriched Semantic Network. In *Proceedings of the Workshop on Applications of WordNet in Natural Language Processing Systems (ACL/COLING 1998)*, 1998.

5. S. Bergamaschi, S. Castano, and M. Vincini. Semantic Integration of Semistructured and Structured Data Sources. *SIGMOD Record*, 28(1):54–59, 1999.
6. 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.
7. 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.
8. C. Bizer. D2R MAP - A Database to RDF Mapping Language. In *Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, 2003.
9. V. Christophides, G. Karvounarakis, I. Koffina, G. Kokkinidis, A. Magkanaraki, D. Plexousakis, G. Serfiotis, and V. Tannen. The ICS-FORTH SWIM: A Powerful Semantic Web Integration Middleware. In *SWDB 2003*, pages 381–393, 2003.
10. I. F. Cruz, H. Xiao, and F. Hsu. Peer-to-Peer Semantic Integration of XML and RDF Data Sources. In *Third International Workshop on Agents and Peer-to-Peer Computing (AP2PC 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. G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: a declarative query language for RDF. In *Proceedings of the 11th International World Wide Web Conference (WWW 2002)*, pages 592–603, 2002.
13. M. C. A. Klein. Interpreting XML Documents via an RDF Schema Ontology. In *Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)*, pages 889–894, 2002.
14. P. Lehti and P. Fankhauser. XML Data Integration with OWL: Experiences and Challenges. In *2004 Symposium on Applications and the Internet (SAINT 2004)*, pages 160–170, 2004.
15. D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technical Report HPL-2002-57, HP Laboratories Palo Alto, 2002.
16. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: A P2P Networking Infrastructure Based on RDF. In *Proceedings of the 11th International World Wide Web Conference (WWW 2002)*, 2002.
17. 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.
18. B. Omelayenko. RDFT: A Mapping Meta-Ontology for Web Service Integration. In *Knowledge Transformation for the Semantic Web 2003*, pages 137–153, 2003.
19. E. Rahm and P. A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB J.*, 10(4):334–350, 2001.
20. 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.