Ontology-based Query Rewriting in Peer-to-Peer Networks

Huiyong Xiao
ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
Chicago, IL
hxiao@cs.uic.edu

Isabel F. Cruz
ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
Chicago, IL
ifc@cs.uic.edu

***Abstract*—We describe an ontology-based approach to the problem of data interoperability, specifically focusing on the issue of query processing in a peer-to-peer (P2P) heterogeneous setting. In particular, we discuss the mechanisms for metadata representation, P2P ontology mapping, and query answering. Our contributions include PML, a P2P mapping language, RDFMS, a meta-ontology for mapping representation, and a query rewriting algorithm that considers integrity constraints specified on local data sources.**

I. Introduction

Research on peer-to-peer (P2P) computing techniques is flourishing with a number of proposals on the related open issues such as robustness in dynamic P2P systems, reliability of participants (peers), network performance, data coordination, and semantic issues [20]. Among these issues, data interoperability is fundamental, especially in the case of fine-grained (e.g., content-based) searches in a P2P network of data sources. Thus, *P2P data management (or integration) systems* (PDMS) arise by combining schema-based data integration with a P2P infrastructure [3, 14]. In addition, the use of ontologies has been recognized as an effective approach to promote the interoperability among distributed sources, by resolving their data heterogeneities at a semantic level [8, 15, 21, 24]. These two research trends lead to the emergence of *ontology-based P2P data management systems* (OPDMS).

P2P ontology mapping and query processing are two important issues in an OPDMS. While ontologies are used in local sources as a uniform conceptual metadata representation, which resolves the syntactic heterogeneity among sources in different peers, schematic (or structural) and semantic heterogeneity may still exist. Therefore, ontology mappings are established between peers to provide a common understanding of their data sources [15]. Based on such ontology mappings, a variety of data management tasks, such as data integration, query processing, and data exchange, can be performed within the whole OPDMS. In this paper, we propose a framework for OPDMS and discuss the issue of query processing in this framework. In particular, we propose a P2P query rewriting algorithm that takes into account integrity constraints specified on local data sources.

In our work, local RDFS[1] ontologies are used to uniformly represent heterogeneous source schemas. To represent the semantic mappings among these metadata (ontologies), we propose a mapping language, namely the P2P Mapping Language (PML), which uses a meta-ontology called RDF Mapping Schema (RDFMS). We also discuss the process of P2P query answering in a layered framework, which we propose to manage any peer. In spite of its simplicity in comparison with some mapping languages (e.g., Semantic Bridging Ontology used in MAFRA [19]), PML is adequately expressive to represent most types of ontology mappings including the *equivalent*, *broader* (more generalized), *narrower* (more specialized), *union*, and *intersection* mappings. Furthermore, PML is extensible to define complex (e.g., many-to-many) mappings and new mapping types (e.g., a *sibling* mapping based on two broader mappings), due to the extensibility of RDFMS as is defined on top of RDFS. We define a first order logic (FOL) semantics for PML, as well as for queries, which lays a unified foundation for query rewriting. We consider a particular class of queries on RDFS ontologies, namely conjunctive RQL (c-RQL) queries, and propose a P2P query rewriting algorithm.

The rest of the paper is organized as follows. In Section II we describe existing related work. Section III gives an overview of our approach. In Section IV, we discuss in detail the P2P mapping language PML, as well as the meta-ontology RDFMS, which is used for mapping representation. The algorithm for P2P query answering, specifically for P2P query rewriting based on the P2P mappings, is given in Section V. Finally, Section VI concludes and discusses future work.

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

II. Related Work

Semantic data integration using conceptual models, such as E-R models and ontologies, has been widely investigated in the literature [18, 21, 24]. Many P2P data management systems (PDMSs) have been recently proposed, such as the LRM model [3] and Piazza [14]. Our framework as proposed in this paper is closer to Piazza, which deals with the integration of XML data and XML serialization of RDF data from different peers. Piazza uses an XQuery-based mapping language to represent schema mappings. Query answering is realized by pattern matching between the tree representing the XQuery and the tree representing the mappings.

Examples of OPDMS include the SWAP architecture [10], and based on it, the Bibster system [12]. Our ontology-based query rewriting algorithm in OPDMS is similar to the computeWTA algorithm proposed by Calvanese *et al.* [6] for query reformulation, both assuming consistent ontology mappings. However, unlike in computeWTA, we allow partial ontology mappings, i.e., it is not necessary to map all the atoms in the query to be rewritten. This assumption is practically meaningful since the user's burden in mapping two peers can be thus reduced.

The representation of ontology mappings should facilitate the use of mappings for data management tasks, including data exchange and query processing. Issues related to ontology mapping have been studied widely [15, 21]. For example, Lehti *et al.* propose an OWL-based model particularly for XML data integration [17]. For representing RDF schema mappings, Omelayenko has proposed the use of a meta-ontology, RDFT [22]. However, it is unclear how execution specific constraint information and data transformation dimension are attached to the bridges. Context OWL (C-OWL) [5] and the Semantic Bridging Ontology (SBO) [19] are two similar ontology mapping languages, with the former based on an extended OWL syntax and semantics and the latter represented in DAML+OIL. Both languages define a set of *bridge rules* with an explicit semantics. However, the utilization of such rules for query processing remains an open issue.

In the case that mappings are defined as (relational) views, query processing is often referred in literature as view-based query answering or rewriting [13]. However, few of view based query processing algorithms address the issue of query writing over ontologies, which usually allows for more expressive constraints specification than most schemata languages do.

III. System Overview

A. The Layered Peer Architecture

In a P2P data management system, a peer manages its local data source as in a traditional database system. In addition, a peer also has to possess the ability of communicating with the other peers by providing and consuming services. To this end, we propose for each peer a layered architecture (as shown in Figure 1), 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 instances. A wrapper is used to convert the local source schemas and data into such local ontologies. The *representation layer* contains the local ontology in RDFS and its mappings in RDFMS. The *service layer* implements schema mapping and query processing, which are two main services that a peer can provide to the network. The *application layer* contains a GUI (Graphic User Interface) for the user to initiate query requests. The adoption of a layered peer architecture simplifies the resolution of peer-to-peer heterogeneities into level-to-level dependencies, thus facilitating the data interoperation by making the layers more maintainable and reusable.

B. An Illustrative Example

As shown in Figure 2, the three autonomous peers p_1, p_2, and p_3 contain three data sources, which are heterogeneous in both syntax and schemata. In particular, Peer p_1 contains the information of faculty and publications in XML (`p.xml`) and DTD (`p.dtd`). The publication element (`pub`) that is defined in IDREFS refers to one or more `publication` IDs (`id`). Such referential constraints define inclusion dependencies as in relational databases. Peer p_2 is a relational database containing conference proceedings. The attributes `aid` and `pid` in `author_proc` are foreign keys referring to `author.aid` and `proceedings.pid`, respectively, defining inclusion dependencies. Peer p_3 contains an RDF document (`f.rdf`)

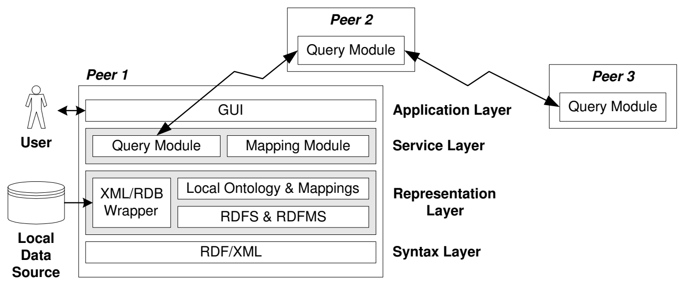

Fig. 1. The layered peer architecture.

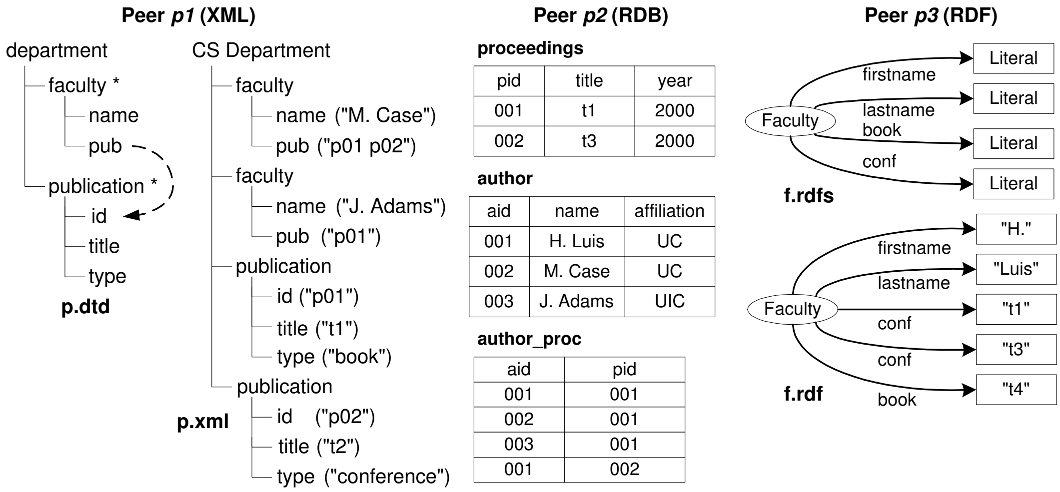

Fig. 2: A motivating example.

with its RDF schema (`f.rdfs`) defined in RDFS. In comparison to XML data, we say that an RDF data is *flat* because there are no nesting structure and order constraints among the classes and properties.

In addition to syntactic heterogeneity, a 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) of an attribute in p_1, as relation names in p_2, and as property names in p_3.

C. RDF Metadata Representation

The source schemas specify metadata about different data sources, in terms of elements and attributes in XML schemas, relations and attributes in relational schemas, and classes and properties in RDFS. A heterogeneous P2P integration system should provide a uniform metadata representation to facilitate the P2P mapping process. For this purpose, wrappers are used to transform heterogeneous schemas into the uniform representation [4, 9, 16].

In our approach, we choose to use RDFS to represent local metadata as a local ontology. The following description summarizes the method of model-based schema transformation in our previous work [9]. For transformation from relational to RDFS, we represent relations as RDF classes and attributes as RDF properties. For transformation from XML to RDF, we convert complex-type elements into RDF classes and simple-type elements (with no subelements but character contents) and attributes into RDF properties. The target RDF schema shall also include the XML or relational *referential constraints*, which are necessary to be preserved for correct query translation between different data sources. We represent these constraints by two RDF properties (corresponding to the two attributes involved in a referential constraint) sharing the same value. Figure 3 shows the results of schema translation of the three sources in the example of Figure 2. Notice that the nesting relationship between two XML elements is preserved by a new particular RDF property `rdfx:contained`, where `rdfx` is the new namespace [8].

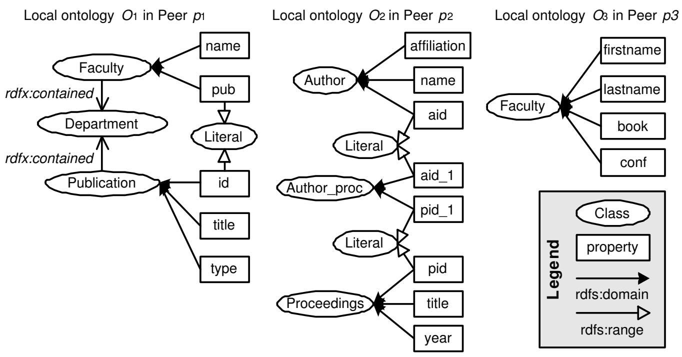

Fig. 3. local RDFS ontologies.

D. P2P Mapping and Query Answering

In our framework, the P2P inter-schema mappings result from a process of matching the two participating source schemas [23]. In our previous work [25], we proposed a thesaurus-based RDF schema matching algorithm by utilizing WordNet.[1] In our approach, an inter-schema mapping specifies correspondences between RDF classes or properties from two different source schemas. The different types of mappings (e.g., *equivalent*, *broader*, or *nar-*

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

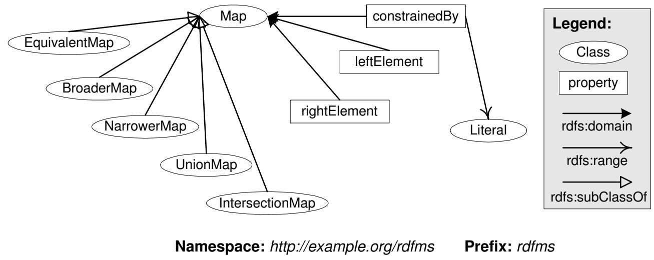

Fig. 4. The meta-ontology of RDFMS.

rower) are determined according to the comparison of the semantics of the mapped classes or properties. The mapping information is stored in terms of instances of an RDF meta-ontology RDFMS (RDF Mapping Schema), using in addition a mapping language, PML (P2P Mapping Language).

The process of P2P query answering includes three aspects: query execution, query rewriting, and answer integration. The user poses a query on a peer, which is first executed on that peer. Meanwhile, the query is also forwarded to each of the linked peers, where the query is rewritten into a new query that is executed locally and propagated further. Finally, answers from every peer are returned to the host peer, where they are integrated to produce the answer. For the purpose of query answering, we use a first-order relation based method to interpret the inter-schema mappings. Actually, in our approach, both the mappings and heterogeneous queries are interpreted by a set of first-order relations, so as to provide a unified environment for query rewriting.

IV. P2P Mappings

P2P semantic mappings between two peers specify how two local schemas are related, thus providing a foundation for P2P query answering. We discuss in this section the issues related to mapping representation and the use of mappings for query processing, instead of describing the detailed mapping process. A thesaurus-based RDF schema matching method can be referred to in our previous work [25]. We also propose an RDF-based meta-ontology RDFMS to P2P semantic mapping representation and discuss its use in a mapping language PML that uses an FOL semantics.

A. RDFMS Meta-ontology

As shown in Figure 4, RDFMS provides one-to-one mappings such as *equivalent* (represented by `EquivalentMap`), *broader* (`BroaderMap`), and *narrower* (`NarrowerMap`). Regarding the case of one-to-many mappings, RDFMS defines `UnionMap` and `IntersectionMap` respectively for two types of logic combinations (i.e., *and* and *or*) of the elements on the multiple-element side. All these 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 [23] that a P2P mapping may carry, the property `constrainedBy` is defined, whose data type is specified as `Literal`. An example of the use of this property is `&c1` (see Figure 5), which is used to confine the retrieval of the instances from Peer p_1 since `Faculty` is mapped to `Author` using `NarrowerMap`. Following the example in Figure 3, we otain the P2P mappings among the three local RDF schemas, as shown in Figure 5. Note that every P2P inter-schema mapping is an instance of the RDFMS meta-ontology.

B. P2P Mapping Language – PML

We define a set of mapping atoms for defining different types of P2P semantic mappings, according to the structure of the RDFMS meta-ontology. Listed below are mapping atoms and their corresponding RDFMS representation.

- $\mathsf{EM}(c_1, c_2)$: there exists an instance m of `EquivalentMap`, such that $c_1 = m.$`leftElement` and $c_2 = m.$`rightElement`.
- $\mathsf{BM}(c_1, c_2)$: there exists an instance m of `BroaderMap`, such that $c_1 = m.$`leftElement` and $c_2 = m.$`rightElement`.
- $\mathsf{NM}(c_1, c_2)$: there exists an instance m of `NarrowerMap`, such that $c_1 = m.$`leftElement` and $c_2 = m.$`rightElement`.
- $\mathsf{UM}(c_1, c_2)$: there exists an instance m of `UnionMap`, such that $c_1 = m.$`leftElement` and $c_2 = \{x|x = m.\texttt{rightElement}\}$, or $c_1 = \{x|x = m.\texttt{rightElement}\}$ and $c_2 = m.$`leftElement`.
- $\mathsf{IM}(c_1, c_2)$: there exists an instance m of `IntersectionMap`, such that $c_1 = m.$`leftElement` and $c_2 = \{x|x = m.\texttt{rightElement}\}$, or $c_1 = \{x|x = m.\texttt{rightElement}\}$ and $c_2 = m.$`leftElement`.
- $\mathsf{CON}(m, e)$: given an instance m of `Map` or its subclasses, we have $e = m.$`constrainedBy`.

We note that c_1 and c_2 in EM, BM, and NM correspond to RDFS classes or properties, whereas c_1 and c_2 in UM and IM can correspond to a set of classes or properties, to which the logic connectors *and* and *or* are applied, respectively.

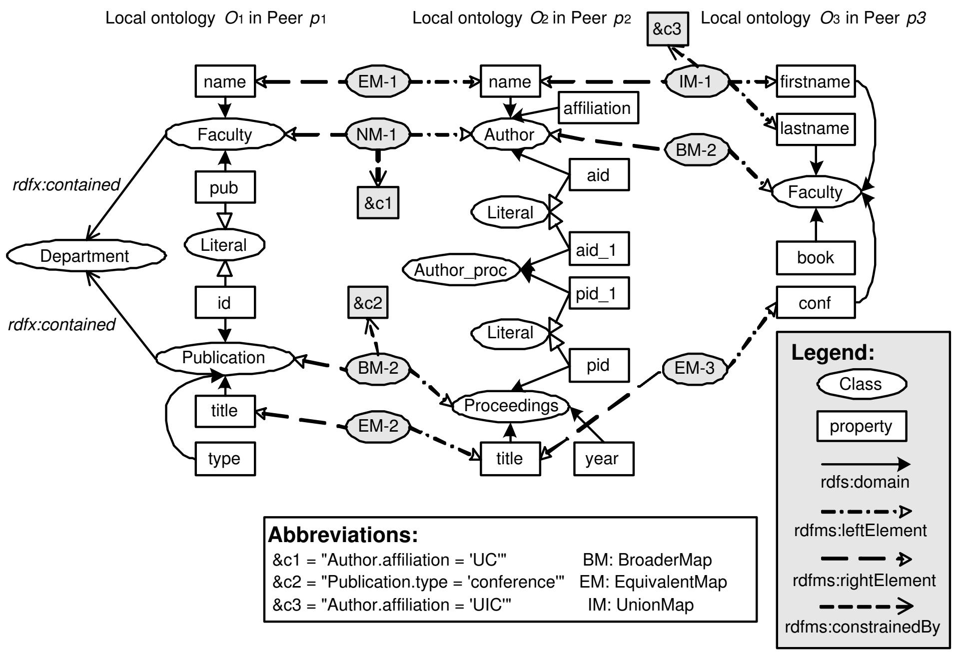

Fig. 5: An example of P2P mappings in RDFMS.

Assuming a finite set of class names $\mathcal{C}$ and a finite set of property names $\mathcal{P}$, we define a FOL (first order logic) semantics for the mapping atoms, in terms of the following two predicates:

- $\mathsf{C\_EXT}(c,\ x)$, where the resource x is in the *proper extent* (i.e., direct instance) of class c.
- $\mathsf{P\_EXT}(x,\ p,\ y)$, where $(x,\ y)$ is the proper extent (i.e., direct instance) of property p.

In our definition, a P2P mapping is allowed to connect not only two classes or two properties but also a class and a property. An interpretation Δ of every P2P mapping atoms varies according to the type of objects that are mapped, as given below.

- $\Delta\mathsf{EM}(c_1,\ c_2)$ implies:

$\forall x\ \mathsf{C\_EXT}(c_1,\ x) \leftrightarrow \mathsf{C\_EXT}(c_2,\ x)$, if $c_1, c_2 \in \mathcal{C}$;
$\forall x_1 \forall x_2 \forall y\ \mathsf{P\_EXT}(x_1,\ c_1,\ y) \leftrightarrow \mathsf{P\_EXT}(x_2,\ c_2,\ y)$, if $c_1, c_2 \in \mathcal{P}$;
$\forall x \forall y\ \mathsf{C\_EXT}(c_1,\ y) \leftrightarrow \mathsf{P\_EXT}(x,\ c_2,\ y)$, if $c_1 \in \mathcal{P}, c_2 \in \mathcal{C}$.

- $\Delta\mathsf{BM}(c_1,\ c_2)$ implies:

$\forall x\ \mathsf{C\_EXT}(c_1,\ x) \rightarrow \mathsf{C\_EXT}(c_2,\ x)$, if $c_1, c_2 \in \mathcal{C}$;
$\forall x_1 \forall x_2 \forall y\ \mathsf{P\_EXT}(x_1,\ c_1,\ y) \rightarrow \mathsf{P\_EXT}(x_2,\ c_2,\ y)$, if $c_1, c_2 \in \mathcal{P}$;
$\forall x \forall y\ \mathsf{C\_EXT}(c_1,\ y) \rightarrow \mathsf{P\_EXT}(x,\ c_2,\ y)$, if $c_1 \in \mathcal{P}, c_2 \in \mathcal{C}$.

- $\Delta\mathsf{NM}(c_1,\ c_2)$ implies:

$\forall x\ \mathsf{C\_EXT}(c_1,\ x) \leftarrow \mathsf{C\_EXT}(c_2,\ x)$, if $c_1, c_2 \in \mathcal{C}$;
$\forall x_1 \forall x_2 \forall y\ \mathsf{P\_EXT}(x_1,\ c_1,\ y) \leftarrow \mathsf{P\_EXT}(x_2,\ c_2,\ y)$, if $c_1, c_2 \in \mathcal{P}$;
$\forall x \forall y\ \mathsf{C\_EXT}(c_1,\ y) \leftarrow \mathsf{P\_EXT}(x,\ c_2,\ y)$, if $c_1 \in \mathcal{P}, c_2 \in \mathcal{C}$.

- $\Delta\mathsf{UM}(c_1,\ c_2)$ implies:

$\bigvee_i(\Delta\mathsf{EM}(c_1,\ a_i))$, where $a_i \in c_2$, if $c_1 \in \mathcal{C} \cup \mathcal{P}$, $c_2 \subseteq \mathcal{C} \cup \mathcal{P}$;
$\bigvee_i(\Delta\mathsf{EM}(a_i,\ c_2))$, where $a_i \in c_1$, if $c_1 \subseteq \mathcal{C} \cup \mathcal{P}$, $c_2 \in \mathcal{C} \cup \mathcal{P}$.

- $\Delta\mathsf{IM}(c_1,\ c_2)$ implies:

$\bigwedge_i(\Delta\mathsf{EM}(c_1,\ a_i))$, where $a_i \in c_2$, if $c_1 \in \mathcal{C} \cup \mathcal{P}$, $c_2 \subseteq \mathcal{C} \cup \mathcal{P}$;
$\bigwedge_i(\Delta\mathsf{EM}(a_i,\ c_2))$, where $a_i \in c_1$, if $c_1 \subseteq \mathcal{C} \cup \mathcal{P}$, $c_2 \in \mathcal{C} \cup \mathcal{P}$.

The following is the interpretation $\Delta M_{1,2}$ for the mappings $M_{1,2}$ between p_1 and p_2 in Figure 5.

$$\forall x_1 \forall x_2 \forall y\ \mathsf{P\_EXT}(x_1, name, y) \leftrightarrow \mathsf{P\_EXT}(x_2, name, y),$$
$$\forall x_1 \forall x_2 \forall y',\ \mathsf{P\_EXT}(x_1, title, y') \leftrightarrow \mathsf{P\_EXT}(x_1, title, y'),$$
$$\forall x\ \mathsf{C\_EXT}(Faculty, x) \rightarrow \mathsf{C\_EXT}(Author, x),$$
$$\forall x\ \mathsf{C\_EXT}(Publication, x) \leftarrow \mathsf{C\_EXT}(Proceedings, x)$$

The FOL interpretation for ontology mappings enables standard reasoning on mappings as well as the definition of more complex P2P mappings. For example, we can define a *sibling* mapping SM such that $\mathsf{SM}(c_1,\ c_2) \Leftrightarrow \mathsf{NM}(c_1,\ c_3) \wedge \mathsf{NM}(c_2,\ c_3)$. Another example is the definition of a many-to-many mapping by composing two `UnionMaps`. Furthermore, an example for reasoning on mappings can be such as $\Delta\mathsf{BM}(c_1,\ c_2) \wedge \Delta\mathsf{NM}(c_1,\ c_2) \Leftrightarrow \Delta\mathsf{EM}(c_1,\ c_2)$. However, reasoning on mappings is not the focus of the this paper. Instead, we concentrate on how to use mappings for the purpose of query processing, specifically on query rewriting.

V. P2P Query Processing

A. Query Languages

Since the metadata of every source schema is expressed as a local ontology in RDFS, we may be able to interpret a local query over the source schema in terms of a conjunctive query, namely a *conjunctive RQL query* (c-RQL) [7], over the local ontology. An c-RQL Q is of the form

$$ans(\mathbf{x}) \coloneq R_1(\mathbf{x}_1), ..., R_n(\mathbf{x}_n).$$

where R_i is either $\mathsf{C\_EXT}$ or $\mathsf{P\_EXT}$ for $i \in [1..n]$, and $\mathbf{x} \subseteq \mathbf{x}_1 \cup ... \cup \mathbf{x}_n$. As usual, the *ans* part is called the head of the query, denoted $head_Q$, and the rest is called the body of the query, denoted $body_Q$. In this paper, we assume that we only consider the class of local queries that can be expressed in c-RQL. The following gives two examples of translating local XPath [11] and relational queries into c-RQL queries, while ignoring the detailed procedure for the space limit.

Consider an XPath query `/department/faculty[@name="M. Case"]` as posed over `p.xml` in p_1. The result of this query is the XML document tree (referred to as *answer tree*) rooted from the first `faculty` element (See Figure 2). By considering the answer structure and semantics of the query (for correct query rewriting), we can interpret the XPath query as follows. Note that all the elements and/or attributes involved in the answer tree and in the predicates (of an XPath query) are covered in the resulting c-RQL query.

$$ans(x,y,z) \coloneq \mathsf{P\_EXT}(x, name, y),\ \mathsf{P\_EXT}(x, pub, z),\ y = \texttt{"M. Case"}.$$

As another example, consider a relational conjunctive query posed on Peer p_1 to "*find all the publications written by authors from UIC*", as shown below.

$$ans(y) \coloneq proceedings(x,y,z),\ authorproc(u,x),\ author(u,v,w),\ w = \texttt{"UIC"}.$$

The following is the first-order relation based interpretation for the preceding relational conjunctive query.

$$\begin{aligned} ans(y) \coloneq\ & \mathsf{P\_EXT}(\hat{x_1}, pid, x),\ \mathsf{P\_EXT}(\hat{x_1}, title, y),\\ & \mathsf{P\_EXT}(\hat{x_2}, aid\_1, u),\ \mathsf{P\_EXT}(\hat{x_2}, pid\_1, x),\\ & \mathsf{P\_EXT}(\hat{x_3}, aid, u),\ \mathsf{P\_EXT}(\hat{x_3}, affiliation, w),\\ & w{=}\texttt{"UIC"} \end{aligned}$$

B. Query Rewriting

The P2P query answering in our framework is a process of propagating a *local query* (initiated from a *host peer* p_1) to every connected peer along the links. As previously mentioned, this process includes three aspects: query execution, query rewriting, and answer integration. Query rewriting can be seen as a function $Q_2 = f(Q_1, M)$, where Q_1 is the local query, M is the set of P2P mappings, and Q_2 is the resulting remote query. Based on the uniform first order logic interpretation for both P2P mappings and user queries, the computation of f is realized by the algorithm P2PRewriting as sketched below.

```
Algorithm P2PRewriting (Q, M)
Input: a conjunctive query Q over ontology O_1; the
mappings M between ontology O_1 and O_2.
Output: a conjunctive query Q' over O_2.
---------------------------------------------------------
head_Q' = head_Q; body_Q' = null;
Let ΔQ be the corresponding c-RQL of Q;
Expand ΔQ into Q* using the constraints over O_1;
Let φ be body_Q*;
For each R(x) of φ
   For each ψ ∈ M
      Let R'(x') be the result of applying ψ on R(x);
      Add R'(x') into body_Q' using a conjunction;
Let G be a query graph of φ and G' be of body_Q';
For each connected subgraph H ⊆ G
   Find the corresponding subgraph H' of H in G';
   If H' is not connected then
      Expand H' using the constraints on O_2 into a
      connected graph H'';
      If H'' exists then add into body_Q' all R'_i that
      contributes to the expansion of H';
      Else output null;
Output Q';
```

The rest of our discussion elaborates on this algorithm by giving a concrete example. Suppose that the user poses a query Q over Peer p_1 (in a P2P network as shown in Figure 3): "*listing all papers written by H. Luis*", which is formulated as follows:

```
//publication[//faculty[contains(@pub, @id) and
            @name="H. Luis"]]
```

The first step of rewriting Q is the interpretation of Q as ΔQ. As previously mentioned, the interpretation of an XPath query has to consider its answer structure. In this example, the answer to Q covers the XML node `publication` and its children `id`, `title`, and `type`, according to the schema structure in p_1 (see Figure 2). Based on the local RDFS ontology of Peer p_1, ΔQ is computed as follows, .

$$\begin{aligned} ans(x,y,z) \coloneq\ & \mathsf{P\_EXT}(p, id, x),\ \mathsf{P\_EXT}(p, title, y),\\ & \mathsf{P\_EXT}(p, type, z),\ \mathsf{P\_EXT}(q, pub, x),\\ & \mathsf{P\_EXT}(q, name, \texttt{"H. Luis"}) \end{aligned}$$

The expansion of ΔQ uses the classic chase algorithm that "chases" a tableau query with dependencies on a relational database [1]. The following shows the resulting Q^* of expanding ΔQ using the constraints on the ontology O_1, and its rewriting Q' (after the application of the mapping constraints $M_{1,2}$).

$$Q^* : ans(x, y, z) \text{ :- } \mathsf{P\_EXT}(p, id, x),\ \mathsf{P\_EXT}(p, title, y),\ \mathsf{P\_EXT}(p, type, z),\ \mathsf{C\_EXT}(Publication, p),\ \mathsf{P\_EXT}(q, pub, x),\ \mathsf{P\_EXT}(q, name, \texttt{"H. Luis"}),\ \mathsf{C\_EXT}(Faculty, q)$$

$$Q' : ans(y) \text{ :- } \mathsf{P\_EXT}(p, title, y),\ \mathsf{C\_EXT}(Publication, p),\ \mathsf{P\_EXT}(q, name, \texttt{"H. Luis"})$$

In the last step, the algorithm finds that the *query graph* (computed by adding a node for each atom and adding an edge between two nodes if their corresponding atoms contain the same variable) of Q^* is connected, whereas the one of Q' is not connected. Hence, Q' is expanded according to the constraints on O_2, resulting in the following final rewriting Q' of Q:

$$ans(y) \text{ :- } \mathsf{P\_EXT}(p, title, y),\ \mathsf{C\_EXT}(Publication, p),\ \mathsf{P\_EXT}(q, name, \texttt{"H. Luis"}),\ \mathsf{P\_EXT}(p, pid, y_2),\ \mathsf{P\_EXT}(q, aid, y_1),\ \mathsf{P\_EXT}(x, aid\_1, y_1),\ \mathsf{P\_EXT}(x, pid\_1, y_2)$$

It will not be difficult to obtain the corresponding relational conjunctive query of Q', which is then executed over the RDB in Peer p_2 to retrieve a local answer from p_2. Similarly, we can rewrite Q' to a query Q'' over O_3 and get a local answer from p_3. The (global) answer of Q, after the integration of all local answers, is as follows, where the null values are caused by the fact that the P2P mappings may be partial:

```
<publication id="" title="t1" type=""/>
<publication id="" title="t3" type=""/>
<publication id="" title="t4" type=""/>
```

We did not describe all the details because of space limitations.

In order to retrieve correct data from the P2P network, it is required that the remote query Q_2 rewritten from a local query Q_1 be equivalent to Q_1. The query rewriting satisfying this condition is called *equivalent query rewriting* [13], which is defined for homogeneous relational data integration. *Query equivalence* in terms of *answer equivalence* has also been defined [13]. Such equivalence, however, will have a different (less strict) meaning in the context of a heterogeneous P2P network. Informally, we say that two answers (to two different queries on different data sources) are equivalent if they are structurally and semantically equivalent. However, such equivalence does not entail identical answers. Although not proved in this paper, our P2P query rewriting guarantees semantic equivalence, which is based on the concept of reversibility [8]. To achieve semantic equivalence the following is needed: the correctness of source schema representation in RDFS, a valid P2P ontology mapping, and the preservation of the answer structures.

VI. Conclusions and Future Work

In this paper, we describe an ontology-based approach to the data interoperability problem in a heterogeneous P2P network. RDF techniques are used in our framework, through the use of the RDFS local ontologies for metadata representation and the use of the RDFMS meta-ontology for inter-schema mapping representation. Our contributions include a definition of the syntax of PML (based on the RDFMS meta-ontology), a definition for its semantics in terms of first-order relations, and a query answering algorithm that considers constraints in local data sources.

For future work, we will further study the following aspects: (1) Due to the locality of P2P systems, mappings between different pairs of peers may be designated by different people. This can result in *inconsistency* between different inter-schema mappings. In addition, given two large-size source schemas to be mapped, the user may hope some inferencing can be performed to derive new mappings from existing mappings automatically. In fact, the problem of *mapping consistency* and that of *mapping inference* are essentially the same in the case where the inferencing involves multiple sets of inter-schema mappings [2]. (2) In a P2P network, peers are designed as autonomous nodes, and any peer can accept user queries. In such settings, an established inter-schema mapping, say from Peer p_1 to Peer p_2, may be used both for query rewriting from p_1 to p_2 and for that from p_2 to p_1. Given that the inter-schema mappings are directional and a uniform query rewriting algorithm is deployed in the P2P system, the utilization of a single inter-schema mapping for query rewriting in different directions have to be treated differently. This arises because of the problem of *bidirectionality* of P2P mappings [14].

VII. Acknowledgments

This research was partially supported by the National Science Foundation under Awards ITR IIS-0326284 and IIS-0513553.

VIII. References

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] Marcelo Arenas, Vasiliki Kantere, Anastasios Kementsietsidis, Iluju Kiringa, Renée J. Miller, and John Mylopoulos. The Hyperion Project: From Data Integration to Data Coordination. *SIGMOD Record*, 32(3):53–38, 2003.

[3] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini,

and Ilya Zaihrayeu. Data Management for Peer-to-Peer Computing: A Vision. In *WebDB 2002*, pages 89–94, 2002.

[4] Christian Bizer. D2R MAP - A Database to RDF Mapping Language. In *Proc. of WWW 2003*, 2003.

[5] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-OWL: Contextualizing Ontologies. In *Proc. of ISWC 2003*, pages 164–179, 2003.

[6] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. What to Ask to a Peer: Ontolgoy-based Query Reformulation. In *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 469–478, 2004.

[7] Vassilis Christophides, Gregory Karvounarakis, I. Koffina, Giorgos Kokkinidis, Aimilia Magkanaraki, Dimitris Plexousakis, G. Serfiotis, and Val Tannen. The ICS-FORTH SWIM: A Powerful Semantic Web Integration Middleware. In *SWDB 2003*, pages 381–393, 2003.

[8] Isabel F. Cruz, Huiyong Xiao, and Feihong Hsu. An Ontology-based Framework for Semantic Interoperability between XML Sources. In *Proc. of IDEAS*, pages 217–226, July 2004.

[9] Isabel F. Cruz, Huiyong Xiao, and Feihong Hsu. Peer-to-Peer Semantic Integration of XML and RDF Data Sources. In *Prof. of AP2PC*, 2004.

[10] Marc Ehrig, Christoph Tempich, Jeen Broekstra, Frank van Harmelen, Marta Sabou, Ronny Siebes, Steffen Staab, and Heiner Stuckenschmidt. SWAP - Ontology-based Knowledge Management with Peer-to-Peer Technology. In *Prof. of WOW 2003*, 2003.

[11] Mary Fernández, Ashok Malhotra, Jonathan Marsh, Marton Nagy, and Norman Walsh. XQuery 1.0 and XPath 2.0 Data Model. http://www.w3.org/TR/xpath-datamodel, W3C Working Draft, October 2004.

[12] Peter Haase, Jeen Broekstra, Marc Ehrig, Maarten Menken, Peter Mika, Mariusz Olko, Michal Plechawski, Pawel Pyszlak, Björn Schnizler, Ronny Siebes, Steffen Staab, and Christoph Tempich. Bibster - A Semantics-Based Bibliographic Peer-to-Peer System. In *Prof. of ISWC 2004*, pages 122–136, 2004.

[13] Alon Y. Halevy. Answering Queries Using Views: A Survey. *VLDB J.*, 10(4):270–294, 2001.

[14] Alon Y. Halevy, Zachary G. Ives, Peter Mork, and Igor Tatarinov. Piazza: Data Management Infrastructure for Semantic Web Applications. In *Proc. of WWW 2003*, pages 556–567, 2003.

[15] Yannis Kalfoglou and Marco Schorlemmer. Ontology Mapping: the State of the Art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[16] Michel C. A. Klein. Interpreting XML Documents via an RDF Schema Ontology. In *Proc. of DEXA 2002*, pages 889–894, 2002.

[17] Patrick Lehti and Peter Fankhauser. XML Data Integration with OWL: Experiences and Challenges. In *2004 Symposium on Applications and the Internet (SAINT 2004)*, pages 160–170, 2004.

[18] Maurizio Lenzerini. Data Integration: A Theoretical Perspective. In *PODS 2002*, pages 233–246, 2002.

[19] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. MAFRA - A MApping FRAmework for Distributed Ontologies. In *Proc. of EKAW 2002*, pages 235–250, 2002.

[20] Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-Peer Computing. Technical Report HPL-2002-57, HP Laboratories Palo Alto, 2002.

[21] Natalya Fridman Noy. Semantic Integration: A Survey Of Ontology-Based Approaches. *SIGMOD Record*, 33(4):65–70, 2004.

[22] Borys Omelayenko. RDFT: A Mapping Meta-Ontology for Web Service Integration. In *Knowledge Transformation for the Semantic Web 2003*, pages 137–153, 2003.

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

[24] Holger Wache, Thomas Vögele, Ubbo Visser, Heiner Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-Based Integration of Information - A Survey of Existing Approaches. In *Proc. of the IJCAI-01 Workshop on Ontologies and Information Sharing*, 2001.

[25] Huiyong Xiao and Isabel 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.