Semantic integration of relational data using SPARQL*

Jinpeng Wang, Zhuang Miao, Yafei Zhang, Jianjiang Lu
Institute of Command Automation,
PLA University of Science and Technology, Nanjing 210007, China
E-mail: wangjinpeng1982@gmail.com

Abstract

Data integration is the main problem encountered by applications that need to query across multiple autonomous and heterogeneous data sources. This paper addresses this problem using logic-based approach. We present a semantic integration infrastructure for relational data. In this integration infrastructure, ontology is used as the mediated schema. The formal semantics of SPARQL is defined according to the W3C Candidate Recommendation and translation algorithm from SPARQL to Datalog is provided. A query rewriting algorithm based on Datalog is also provided for heterogeneous data integration.
Key words: Data Integration; SPARQL; Datalog; Query Rewriting

1. Introduction

Integrating and querying data from heterogeneous sources is a hot research topic in database. The goal of data integration is to provide user a uniform access to multiple heterogeneous data sources. In order to achieve this goal, the meaning of the source schemas has to be understood. Being an "explicit specification of a conceptualization"[1], ontology is considered as a possible solution to represent the content of heterogeneous data sources. In our proposal, ontology is used as mediated schema for the explicit description of the data source semantics, providing a shared vocabulary for the specification of the semantics. The relations in the mediated schema are virtual in the sense that their extensions are not actually stored anywhere. The data integration system has a set of source descriptions that specify the semantic mapping between the mediated schema and the source schemas and uses these source descriptions to reformulate a user query into a query over the source schemas. This problem is known in the literature as query rewriting and query answering using views, and has been studied very actively in the recent years [2]. However, with the use of ontology as mediated schema, these former research works are not applicable. To solve this problem, a logic-based approach is proposed. We adopt and refine a recent proposal to formalize the semantics of SPARQL from [3]. Based on the semantic formalization, we provide a translation algorithm from a large fragment of SPARQL queries to Datalog, according to the translations described in [4]. After representing queries using Datalog, we can execute the MiniCon query rewriting algorithm[5] to convert a user query based on mediated schema to queries based on source schema.

The contributions of the present work are:

- A semantic integration infrastructure for relational data.
- Translation algorithm from SPARQL to Datalog.
- Refinement of the MiniCon query rewriting algorithm.

The remainder of this paper is structured as follows: In Sec. 2 we first list some related work and Sec. 3 overview the integration infra-structure, introduce the architecture of data integration system we present. In Sec. 4 we discuss our approach in detail. After pointing out the need for the formal semantics of SPARQL and then define it (Sec. 4.1), we proceed with the translations of SPARQL to Datalog in Sec. 4.2. We describe and refine the MiniCon algorithm to execute the query rewriting in Sec. 4.3. Sec. 5 concludes the paper.

2. Related work

There are a number of researches addressing the problem of data integration among heterogeneous data sources, which can be reduced to three main sub problems listed below:

2.1. Mediated schema

Mediated schema is the logical schema a data integration system employs for uniform expression among data sources. It is usually accompanied by the definition of semantic mappings between the mediated schema and the source schemas. There are several existing approaches to data integration, like SIMS[6], TSIMMIS[7], OBSERVER[8], Information Manifold[9], etc. They usually take relational or XML schema as their mediated schema because it is easy to establish mapping between mediated schema and source schema. However, data source may be

* This work was supported by the National High Technology Research and Development Program of China (No. 2007AA01Z126, 863 Program)

978-0-7695-3497-8/08 $25.00 © 2008 IEEE
DOI 10.1109/IITA.2008.72

heterogeneous both structurally and semantically; this kind of mediated schema cannot deal with the latter case.

Ontology can be the solution to this challenge. Surveys on ontology-based information integration are presented in [10], [11]. There are already some systems employing ontology as mediated schema[12], which map the relational database schemas into corresponding classes or properties in ontology. But manually establishing semantic mappings is a very difficult and imprecise task.

2.2. Query language

One of the main limitations of traditional data integration has been the incapacity to describe semantic queries with traditional query language (e.g. relational or XML). The recent emergence of The Semantic Web has raised a new opportunity and challenge to this problem. In 2004 the RDF Data Access Working Group released a first public working draft of a query language for RDF, called SPARQL[13]. Currently SPARQL is a W3C Candidate Recommendation. It provides powerful facilities for user to formulate complex semantic query.

2.3. Query rewriting algorithms

The problem of "Answering Queries Using Views" considers how to rewrite a conjunctive query using views. A survey and analysis on different algorithms to solve the problem is given in [2]. An effective algorithm called MiniCon can find the maximally-contained rewriting of a conjunctive query using a set of conjunctive view, and experimental study show that the MiniCon algorithm scales up well and significantly outperforms other algorithms (i.e. bucket algorithm, inverse-rules algorithm)[5]. In our integration architecture, we refine the MiniCon algorithm for query rewriting.

3. Overview of the integration infrastructure

In this section, we discuss the architecture the for data integration. Our approach adopts a so-called mediator-wrapper architecture that allows data sources to function independently while the remote access can be done via a mediator and adaptable wrappers. Illustrated in Figure 1, the architecture of our system may be divided into four layers: user interface layer communicate with users; mediating layer contains a mediator which allows the integration; wrapper layer contains wrappers for each data resource; and source layer contains a set of heterogeneous sources.

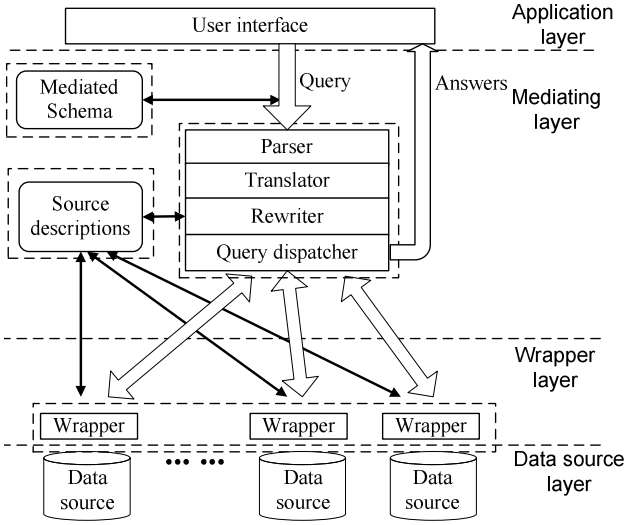

Figure 1 Architecture of data integration system

Circled by broken line, the main elements of the architecture include four parts: query processor, mediated schema, source description and wrapper.

3.1. Query processor

The query processor is the kernel component of data integration system. It parses, translates, rewrites and dispatches the user query to related data sources. When user pose their queries expressed in SPARQL using terms from mediated schema, the parser analyzes the query, verifying if it is in accordance with the SPARQL syntax. Then the translator converts the SPARQL query to equivalent Datalog query, which is the input of rewriter. Rewriter implements the MiniCon algorithm to carry out the query rewriting work with reference to source descriptions.

3.2. Mediated schema

We use ontology as the mediated schema, which can be seen as a knowledge base of a particular domain we are interested. The mediated schema has two roles: (1) It provides the user access to the data with a uniform query interface to facilitate the formulation of a query on all sources; (2) It serves as a shared vocabulary set for wrappers to describe the content in every data sources. The mediated schema is expressed using RDFS in our work.

3.3. Source descriptions

As mentioned before, queries are posed in terms of the mediated schema. To answer a query, the rewriter need descriptions that relate the contents of each data source to the classes, attributes and relations in the mediated schema. Each data source is described by one or more

SPARQL queries. These semantically rich descriptions help the rewriter to form queries and also direct the query dispatcher to distribute queries to specific data sources.

3.4. Wrapper

The wrapper provides an SPARQL view representing a data source and a means to access and to query the data source. It translates the incoming queries into source-specific queries executable by the query processor of the corresponding sources.

4. A logic-based approach for relational data integration

The semantics of SPARQL is still not formally defined in its current version. A recent proposal has tackled this lack [3]. Base on this proposal, as shown in [4] the semantics of SPARQL SELECT queries can, to a large extent, be translated to Datalog programs. Hence, we turn the problem of rewriting SPARQL query into rewriting Datalog query, which has been studied for a long time. In this section, we will exemplify the whole integration procedure of our approach by means of illustrating an example.

Suppose there are three autonomous data source, called source1, source2 and source3. As shown in Figure2, they contain information about authors and papers.

Source1	
PK	**author**
	university

Source2	
PK	**title**
	author conference

Source3	
PK	**paper**
	author published_in works_for

Figure 2 Schema of three data sources

As shown in figure 3, we build an ontology serving as the mediated schema.

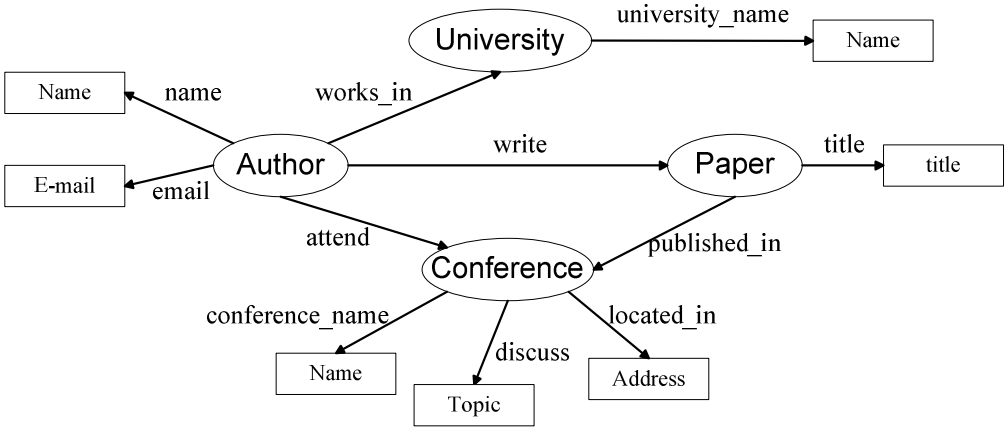

Figure 3 Mediated schema

Translate(V: return variables list, P: graph pattern expression, D: data set, i: counter)

Initialize $i = 1$;

if P is a triple pattern, then return $\text{QUERY}_i(\overline{V},D) :\!- triple(s,p,o,D)$;

if $P = P_1$ AND P_2, then return Translate(vars(P_1), P_1, D, $2i$) $\cup$ Translate(vars(P_2), P_2, D, $2i$+1)

$\cup\ \text{QUERY}_i(\overline{V},D) : -\text{QUERY}_{2i}(\overline{vars(P_1)},D), \text{QUERY}_{2i+1}(\overline{vars(P_2)},D)$;

if $P = P_1$ UNION P_2, then return Translate(vars(P_1), P_1, D, $2i$) $\cup$ Translate(vars(P_2), P_2, D, $2i$+1)

$\cup\ \text{QUERY}_i(\overline{V[V\backslash vars(P_1))\rightarrow \text{null}]},D) : -\text{QUERY}_{2i}(\overline{vars(P_1)},D)$

$\cup\ \text{QUERY}_i(\overline{V[V\backslash vars(P_2))\rightarrow \text{null}]},D) : -\text{QUERY}_{2i+1}(\overline{vars(P_2)},D)$;

if $P = P_1$ MINUS P_2, then return Translate(vars(P_1), P_1, D, $2i$) $\cup$ Translate(vars(P_2), P_2, D, $2i$+1)

$\cup\ \text{QUERY}_i(\overline{V[V\backslash vars(P_1))\rightarrow \text{null}]},D) : -\text{QUERY}_{2i}(\overline{vars(P_1)},D), \text{not QUERY}_{2i}'(\overline{vars(P_1)},D)$

$\cup\ \text{QUERY}_{2i}'(\overline{vars(P_1)},D) : -\text{QUERY}_{2i}(\overline{vars(P_1)},D), \text{QUERY}_{2i+1}(\overline{vars(P_2)},D)$;

if $P = P_1$ OPT P_2, then return Translate(V, (P_1 AND P_2), D, i) $\cup$ Translate(V, (P_1 MINUS P_2), D, i);

Figure 4 the translation algorithm from SPARQL queries to Datalog

4.1. Formal semantic of SPARQL

Despite being in the Last Call stage of the W3C recommendation track, the SPARQL query language document currently lacks mathematical rigor and fails to accurately define the semantics for some cases. Pérez introduce the formalization of SPARQL semantic in [3].

4.2. Translation from SPARQL to Datalog

In figure 4 we present a translation algorithm from SPARQL to Datalog following the approach in [4].

As mentioned above, data sources can be modeled as views of mediated schema. We define these views using SPARQL. Three data sources shown in figure 2 can be described as the following SPARQL queries:

source1:

```
SELECT ?author ?university
WHERE
  {  ?X name ?author.
     ?X works_in ?Y.
     ?Y university_name ?university }
```

source 2:

```
SELECT ?title ?author ?conference
WHERE
  {  ?X title ?title.
     ?X published_in ?Z.
     ?Y write ?X.
     ?Y name ?author.
     ?Z conference_name ?conference }
```

source 3:

```
SELECT ?paper ?author ?published_in ?works_for
WHERE
{  ?X title ?paper.
   ?X published_in ?Z.
   ?Y write ?X.
   ?Y name ?author.
   ?Y works_in ?U
   ?Z conference_name ? published_in.
   ?U university_name ?works_for }
```

After the translation, we get the following Datalog queries:

```
SOURCE1(AUTHOR, UNIVERSITY)
        :- (X name AUTHOR),
        (X works_in Y),
        (Y university_name UNIVERSITY).
SOURCE2(PAPER, AUTHOR, CONFERENCE)
        :- (X title PAPER),
        (X published_in Z),
        (Y write X),
        (Y name AUTHOR),
        (Z conference_name CONFERENCE).
SOURCE3(PAPER, AUTHOR, CONFERENCE, UNIVERSITY)
        :- (X title PAPER),
        (X works_in U),
        (X published_in Z),
        (Y write X),
        (Y name AUTHOR),
        (Z conference_name CONFERENCE),
        (U uinversity_name UNIVERSITY).
```

FormMCD(Q, V)
/*Q is a conjunctive query, $Q = \{q_1, \ldots, q_m\}$, V is a set of views, $V = \{V_1, \ldots, V_n\}$
each $V_j (1 \leqslant j \leqslant n)$ is a conjunctive query, $V_j = \{v_{j1}, \ldots, v_{jp}\}$*/
Initialize $M = \Phi$;
for each $q_i \in Q$, $(1 \leqslant i \leqslant m)$, do
  for each $V_j \in V$, and each subgoal $v_{jk} \in V_j$, $(1 \leqslant k \leqslant p)$, do
    if exist a mapping τ that map q_i to v_{jk}, then
    find the maximal subset (denoted by θ) of subgaols of Q that satisfy the *property 1*(see [5])
    $M = M \cup < \tau, \theta >$;
  end for
end for
return M;

Figure 5 forming the MCDs

CombineMCDs(Q, M, A)
/*Q is a conjunctive query, $Q = \{q_1, \ldots, q_m\}$,
M is a set of MCDs, $M = \{m_1, \ldots, m_n\}$
A is a set of rewritings*/
Initialize $Q' = \Phi$;
for each minimal subset $\{m_1, \ldots, m_k\}$ of M such that
  $\theta_1 \cup \theta_2 \cup \ldots \cup \theta_k = Q$
  Create the conjunctive rewriting Q' contain all views in $\{m_1, \ldots, m_k\}$
  Add Q' to A
end for
return A

Figure 6 combining the MCDs

4.3. Query rewriting

As shown in figure 5 and figure 6, we modify the MiniCon algorithm for query rewriting. MiniCon is an effective query rewriting algorithm for data integration. This algorithm has two phases: first, finding the mapping information, which is called MiniCon Description (MCD), between views and the subgoals of query; second, combining the MCDs to form conjunctive rewritings. With the help of MCDs, the MiniCon algorithm can drastically reduces the search space of answers. However, rewriting will contain many redundant subgoals because the relationship between MCDs is not considered. Here

we modify the MiniCon algorithm to get better performance.

Consider the following query, asking for titles and authors of papers written by faculty from PLAUST University. This query is posed as the following SPARQL query:

```
SELECT ?title ?author
WHERE {  ?X name ?author.
         ?X write ?Y.
         ?X works_in ?Z.
         ?Z university_name "PLAUST".
         ?Y title ?title. }
```

This can be translated to the following Datalog query:

```
QUERY(TITLE, AUTHOR)
        :- (X name AUTHOR),
           (X write Y),
           (X works_in Z),
           (Z university_name "PLAUST"),
           (Y title TITLE).
```

After the query rewriting, we get the following two conjunctive queries:

```
QUERY1(TITLE, AUTHOR)
   :- SOURCE1(AUTHOR, UNIVERSITY),
      SOURCE2(TITLE, AUTHOR, CONFERENCE),
      UNIVERSITY = "PLAUST"
QUERY2(TITLE, AUTHOR)
   :- SOURCE3(TITLE, AUTHOR, CONFERENCE, UNIVERSITY),
     UNIVERSITY = "PLAUST"
```

5. Conclusions and future work

The integration of data from multiple heterogeneous sources is an old and well-known research problem for the database and AI research communities. In this paper we considered the problem of integrating heterogeneous relational data sources using SPARQL. We discussed the main issues and also solutions. A novel data integrating architecture based on logic was presented. We used ontology as the mediated schema for integration. Heterogeneous relation schema was model using views defined by SPARQL and translated into Datalog. Following the formalization and translation of SPARQL in [4], we present a translation algorithm from SPARQL to Datalog. The MiniCon algorithm was modified for our integration.

The architecture and approach we provided in this paper can be extended for other data source (i.e. XML, RDF, web forms, etc.); some of the complex semantic (i.e. CONSTURCT, ASK) are not discussed. We leave these problems for future work.

References

[1] Thomas R Gruber (1993). "A translation approach to portable ontology specifications". *Knowledge Acquisition*. Vol.5, No.2, pp. 199-220.

[2] Alon Y Halevy (2001). "Answering queries using views: a survey". *In: VLDB Journal*. Vol.10, No.4, pp. 270-294.

[3] Pérez J, Arenas M, Gutierrez C (2006). "Semantics and complexity of SPARQL". *In: International Semantic Web Conference (ISWC 2006)*. pp. 30–43.

[4] Polleres (2007). "From SPARQL to rules (and back)". *In: Proceedings of the 16th World Wide Web Conference (WWW2007)*. pp. 787-796.

[5] Rachel Pottinger, Alon Halevy (2001). "MiniCon: A scalable algorithm for answering queries using views". *In: VLDB Journal*. Vol.10, No.2-3, pp. 182-198.

[6] Yigal Arens, Craig A Knoblock, Wei-min Shen (1996). "Query reformulation for dynamic information integration". *Juournal of Intelligent Information Systems*. Vol.6, No.2-3, pp. 99-130.

[7] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, et al (1994). "The TSIMMIS Project: Integration of Heterogeneous Information Sources". *Journal of Intelligent Information Systems*. pp. 7-18.

[8] Eduardo Mena, Arantza Illarramendi, Vipul Kashyap, et al (1996). "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies". *In: Proceedings of the First IFCIS International Conference on Cooperative Information Systems*. Vol.8, No.2, pp. 223-271.

[9] Jeffrey D Ullman (1997). "Information integration using logical views". *In: Proceedings of the 6th International Conference on Database Theory*. pp. 19-40.

[10] Noy N F (2004). "Semantic integration: a survey of ontology-based approaches". *In: ACM SIGMOD Record*. Vol.33, No.4, pp. 65-70.

[11] Wache H, Vögele T, Visser U, et al (2001). "Ontology-based integration of information - a survey of existing approaches". *In: Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*. pp. 108-117.

[12] Huajun Chen, Zhaohui Wu, Heng Wang, Yuxin Mao (2006). "RDF/RDFS-based relational database integration". *In: Proceedings of the 22nd International Conference on Data Engineering*. pp. 94.

[13] Eric Prud'hommeaux, Andy Seaborne. "SPARQL query language for RDF". W3C Recommendation 15 January 2008.