Informative Query Answering by Using RDQL for E-Commerce

Myat Thuzar Tun, Soe Yu Maw
University of Computer Studies, Yangon
rnn@mail4u.com.mm

Abstract

Today, the Internet can be seen as a global market place populated by a huge number of providers and consumers that exchange data from a wide range of domains. Data querying is a routine practice for many on-line services. Informative query answering may improve on-line shopping service by incorporating data integration techniques. In this paper, we propose an approach for informative query answering using Resource Description Framework Query Language (RDQL) and Resource Description Framework (RDF). The RDF is used to represent all instance information. RDQL queries RDF documents using SQL-alike syntax. And our framework is based on a mediator/ wrapper approach.

Keywords: Resource Description Framework Query Language (RDQL), Resource Description Framework (RDF), mediator, Informative Query Answering

1. Introduction

Nowadays, large modern enterprise has different portions of the organization using different database management systems to store and search critical data. All of these databases are very important for the enterprise and they have different interfaces for their administration. It will be useful for the enterprise to answer data query. Data querying is a routine practice for many on-line service. Solving queries to support e-commerce transactions can involve retrieving and integrating information from multiple information resources.

Intelligent query answering can provide interesting services for e-commerce applications. Informative query answering by providing summary information is one kind of intelligent query answering. Such information can be obtained by using data warehousing and descriptive data mining techniques [9]. In contrast to other approaches, our approach is based on a mediator/wrapper approach instead of data warehousing approach. The notion of mediation has been proposed as the principle means to resolve problems of semantic interoperation [5].Mediator reconcile the varying semantics of the different data sources. Wrappers lifts heterogeneous data source onto a common data model [14]. Currently, there are two important technologies for semantic integration: Extensible Markup Language (XML) and the Resource Description Framework (RDF). Data in semantic format like RDF may be linked to the ontologies described in the same format [17]. RDF is more semantic than XML. RDF allows us to reason concreted about the real world.

2. Related Works

XML supports a textual representation of data by using application-specific tags. However, XML does not enforce a semantically meaningful data exchange. Since different providers can define different tags to represent the same or semantically similar information [18]. RDF has proven to be a very useful way to represent arbitrary forms of metadata for integration [17]. XML has disadvantages when it comes to semantic interoperability [2].

Computational information sources store and provide large amounts of data that accessing, finding or summarizing information remains a difficult task given the sheer amount of information to be found in each source and given the large number and variety of sources available through current technologies, such as the WWW [13].

XML and RDF are the current standards for establishing semantic interoperability on the web, but XML addresses only document structure. RDF better facilitates interoperation because it provides a data model that can be extended to address sophisticated ontology representation techniques [10]. Approaches that extract and store data locally are suitable for data that varies slowly but return stole values for sources that change faster than the warehouse update cycle.

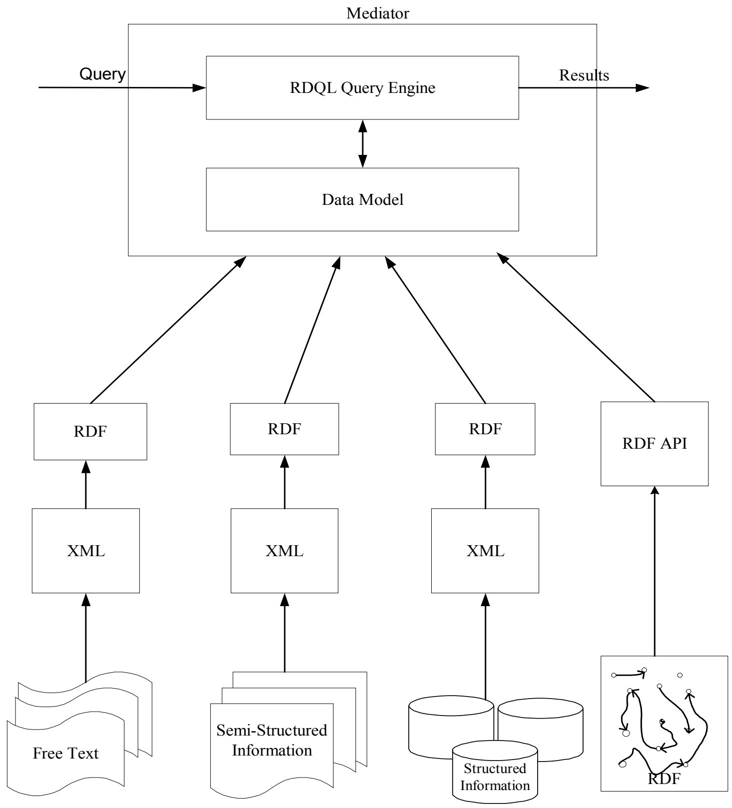

Figure: Proposed System Architecture

3. Overview of proposed design

A diagram overview of proposed design appears in Figure 1. The system consists of the following functional elements: a data model, data wrappers, RDQL query engine and a mediator. Data wrappers, over each piece of source data are used to provide the needed data on a pre-query basis because the breath and depth of queries prevent us from storing or caching all data locally.

The mediator system is to provide unified access to various data sources. The user either sends a query through the user interface or submits an RDQL query. The mediator passes the results back to the user interface.

Mediator

The mediator's responsibility is to propagate information in the query to relevant query modules. Mediators provide unified access to disparate data. The idea is that a query entered by the user is sent to the mediator which sends the query to a variety of sources and combines the results and represent a result set to the user that appears to have come from a single source.

Mediator has two parts.

1 A data model
2 A RDQL query engine

Data Model

This ensures that the query engine is able to identify what information is available without having to access the schema of the resource. The source selection process is aided by data model. The data model consists of information about various data sources as well as relationship between the data source and domain predicates.

RDQL Query Engine

RDQL query engine can be used to evaluate RDQL queries. When query engine receive a query, query engine first decide which resources are relevant to that query. It obtains a list of currently available and relevant resources by consulting the data model. Based on this information, the query engine decomposes the query into sub-queries. Then query engine send the sub-queries to the resources. Once the results are received, the query engine integrates these results.

4. Query Processing

After submitting a query from a user, the query should be routed only to data sources able to answer the query at query time.

Query processing includes three steps as shown in Figure 2:

1. Parsing and translation
2. Optimization
3. Evaluation

4.1. Parsing and Translation

The mediator accepts query in a RDF query language. In this paper we assume RDQL query language. A query consists of two basic parts: The compulsory query part and the mandatory part. The former include constraints and predicates expressed by the SELECT…..WHERE parts and the latter specify first the registry to search for data sources and then mandatory characteristics of the data source. Then it is translated into a machine discipherable concept tree using Boolean algebra.

4.2. Optimization

Query property analyzer analyzes and retrieves the attributes of a query used in the SELECT….WHERE part. The result is send to data model connector.

Data source analyzer and retrieves the mandatory data source characteristics and retrieves URI of data source and forward result to the data model connector.

Data model connector collects the required query and data source characteristics and formulates a query against a given data model.

Data model consists of an ontology describing a set of data source.

Query Rewriter retrieves URIs of data sources matching to the characteristics specified in the mandatory part of a query.

4.3. Evaluation

The query execution engine takes a query evaluation plan by consulting the data model. Based on this plan the query executes and returns the answer to the query.

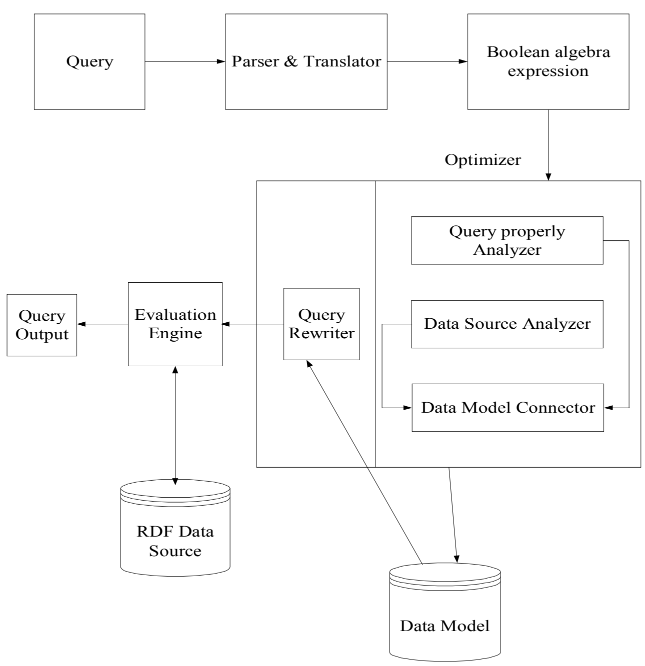

Figure 2: Steps in Query Processing

4.4 Query Example

We use a reference table and RDF description for this to present an RDQL query example.

Table 1: A reference to an RDF description

Brand	Company	Country	Storage-size	Price
MSI	MSI	Japan	128	40.1
Kingstang	Kingstang	Korea	128	39.5
…………				

usb.rdf

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/
1999/02/22-rdf-syntax-ns#"
xmlns:usb="http://www.usbshop.gd
l/usb">
<rdf: Description
 rdf:about="http://www.usbshop.
gdl/usb/MSI">
<usb:company>MSI</usb:company>
<usb:country>Japan
</usb:country>
<usb:memorysize<128
</usb: memorysize>
<USB: price>40.1</usb: price>
</rdf: Description>

<rdf: Description
 rdf:about="http://www.usbshop.
gdl/usb/KingStang">  <usb:
company>KingStang
</usb:company>
<usb:country>Japan
</usb:country>  <usb:
memorysize>128
</usb: memorysize>
<usb: price>39.5</usb: price>
</rdf:Description>
.
.
.
</rdf:RDF>
```

Above usb.rdf describes information about some USB. We indicate the company produced them, country where the company is located, USB's memorysize and price.

Consider the case a user likes to find a USB with the memory size of 128MB and the price of less than $40.

The user, a human, an agent or a semantic web application, issues the following query.

Query:

```
SELECT ?x
FROM <usb.rdf>
WHERE ( ?x, <usb: memorysize>,? z) ,
        (?x, <usb:price>,?y)
AND ?z==128 && ?y<40
USING usb for < http://www.
          usbshop.gdl/usb> ,
      rdf for < http://www.w3.
org/1999/02/22-rdf-syntax-ns#>

Result: Array( [0]->([?x]->http:
//www.usbshop.gdl/usb/MSI).....
```

5. Theoretical Background

5.1 Resource Description Framework (RDF)

An RDF model is graph, often expressed as a set of triples. RDF is a general format for metadata encoding. The idea of RDF is to use metadata to describe the data contained on the web in order to make machine-understandable. The huge amount of data on the web makes it difficult to find, present and maintain the information. It is very hard to automate anything on the web. The semantic of data helps computer for better research and reasoning. RDF defines the rules for defining and expressing the semantic. The big difference between XML and RDF is that XML a labeled tree and RDF a labeled directed graph.

Let me take as an example a single RDF assertion. Let's try "The author of the *page* is *Ora*". This is traditional. In RDF this is a triple

triple (author, page, Ora) which you can think of as represented by the diagram

Figure2: A simple RDF graph

5.2 Resource Description Framework Query Language (RDQL)

RDQL is an implementation of an SQL-alike query language for RDF. It treats RDF as data and provides query with triple patterns and constraints over a single RDF model. While not yet a formal standard, RDQL is widely implemented by RDF frameworks. RDQL allows complex queries to be expressed concisely, with a query engine performing the hard work of accessing the data model. Query provides one way in which the programmer can write a more declarative statement for what is wanted and have the system retrieve it. RDQL's syntax superficially resembles that of SQL, and indeed, some of its concepts will be familiar to anyone who has worked with relational database queries. You can do a lot of thing with RDQL and there're a lot of interesting queries that can be performed.

Features:

1 SQL-like language for retrieving sets of values

2 Java query engine for Jena models

3 Command line support for exploring data sets

4 RDQL queries can be applied to documents in the local file system, URLs of combination.

5 Multiple documents can be queried

An RDQL query contains of the following: (just like SQL query):

1. **SELET** clause- listing all the return values you want
2. **FROM** clause-where to get the information from. In the case of RDQL query, it is the RDF Model from which to make the query. An RDF model could either a file on the file system, and URL that allows the query engine to locate the model, or an in-memory model of the semantic triple.
3. **WHERE** clause- conditions for the query

5.3 Informative query answering

Query answering mechanisms can be classified into two categories based on their method of response: direct query answering and intelligent (or cooperative) query answering. Direct query answering means that a query is answered by returning what is being asked, whereas intelligent query answering consists of analyzing the intent of the query and providing generalized, neighborhood or associated information relevant to the query.

A few example of intelligent query answering are:

1 informative query answering by providing summary information

2 suggestion of additional items based on association analysis

3 Product promotion by sequential pattern mining

Information query answering by providing summary information: When a customer request a regarding a particular book, additional summary information can be provided, such as the volume sold for that book in last year, or attractive feature of it.

6 Conclusions

Our approach has many advantages over the data warehousing approach. Because the system accessed data only in response to user queries, the data is always fresh. The mediation only retrieves data relevant to user queries. It is more scalable than warehouse-based approaches. Data warehousing approaches may return stale values for sources that charge faster than the warehouse update cycle.

Reference:

[1]. Alexander Madche, Steffen Staab and Rudi Studer, "Ontology-Oriented Information Extraction and Integration," *Institute AIFB, Karlsruhe University, 76128 Karlsruhe.*

[2]. Boley Harold, "A web data model unifying XML and RDF," *German Research Center for Artificial Intelligence GmbH, 18.9.2001; available online at http://www.dfki.uni-kl.de/~boley/.*

[3]. Dieter Fensel, Ying Ding, Borys Omelayenko, Ellen Schulten, Guy Botquin, Mike Brown and Alan Flett, "Product Data Integration in B2B E-Commerce," *IEEE INTELLIGENT SYSTEMS, 2001 IEEE.*

[4]. Farshad Hakimpour and Andreas Geppert, " Ontologies: an Approach to Resolve Semantic Heterogeneity in Databases,", *March 8, 2001.*

[5]. G.Wiederhold and Michael R.Genesereth, "The Basic for Mediation," *In CoopIs, Vienna, Austria, May 1995, pp 140-157.*

[6]. Hario Benetti , Domenico Beneventano, Sonia Bergamaschi, Francesco Guerra, and Maurizio Vincini, Universita di Modena e Reggio Emilia. "An Information Integration Framework for E-Commerce," *Universita di Modenae Reggio Emilia. IEEE INTELLIGENT SYSTEMS,2002 IEEE*

[7]. Jaap Gordijn, Hans de Bruin and Hans Akkermans, "Integrate Design of E-commerce Systems: Aligning the Business with software Architecture through Scenerios.

[8]. Jena.sourceforge.net "An Introduction to RDF and the Jena RDF API"

[9]. Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques," *Simon Fraser University.*

[10]. Martin Michalowski, Jose' Luis Ambite, Snehal Thakkar, and Rattapoom Tuchinda," Retrieving and Semantically Integrating Heterogeneous Data from the web," *University of Southern California, IEEE INTELLIGENT SYSTEM, IEEE 2004.*

[11]. M.Saelee, S.Beitzel, E.Jensen, D.Grossman, O.Frieder, "Intranet Mediators: APrototype", *Information Retrieval Laboratory, Department of Computer Sceiences, Illinois Institute of Technology, Proceedings of the International*

Conference on Information Technology: Coding and Computing(ITCC'01, 2001 IEEE.

[12]. Paul Benjamin Lowry," XML Data Mediation and Collaboration: A proposed Comprehensive Architecture and Query Requirements for Using XML to Mediate Heterogeneous Data Sources and Targets," *University of Arizona,2001,IEEE.*

[13]. Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann and Ian Horrocks, "The Semantic Web: The Roles of XML and RDF," *available on-line at http://computer.org/internet/, September-October 2000, IEEE Internet Computing.*

[14]. Steffen Staab," The Semantic Web - New Ways to Present and Integrate Information," *Institute for Applied Informatics and Format Description Methods (AIFB), University of Karlsruhe (TH).*

[15]. S.Staab Michael Erdmann, Alexander Maedche, Stefan Decker "Ontologies in RDF," *AIFB, University of Karlsruhe, D-76128 Darlsruhe, Germany.*

[16]. Suhail Ansari, Ron Kohavi, Llew Mason, and Zijian Zhong,, "Integrating E-Commerce and Data Mining Architecture and Challenges,", *Blue Martini Software, 2600 Campus Drive, San Mateo, CA, 94403, USA.*

[17]. Tom Barret, David Jones, Jun Yuan, John Sawaya, Mike Uschold, Tom Adams, Deborah Folger "RDF Representation of Metada for Semantic Integration of Corporate Information Resources," *The Being Company, MS pp29-99.*

[18]. World Wide Web Consortium, "Extensible Markup Language (XML) 1.0," *Feb.10.1998.*

[19]. www.w3.org, "RDF Primer," W3C Working Draft *March 19, 2002.*

[20]. Zhan Cui, Dean Jones and Paul O' Brien, "Issues in Ontology-based Information Integration," *Intelligent Business System Research Group, Intelligent System Lab, BTexact Technology.*